# Moving the Dialogue Forward:Virtual Student Conversational Agent Design in Low Data Environments

**Maria Phillips**

**Committee Members:**

Professor Donald Brown (Advisor)

Professor Laura Barnes (Chair)

Professor Rafael Alvarado

Professor Afsaneh Doryab

Professor Michael Porter

A dissertation presented for the degree of

Doctor of Philosophy

Engineering Systems and Environment

University of Virginia

July 2022

*This dissertation is dedicated first to God: my source of life, my reason to hope, and my purpose for living. By His grace I am what I am; may my life glorify You.*

*Next, to my beloved, Zachary, a constant source of encouragement during life's challenges. You have met me without judgment in some of my darkest places, providing true love and stable support that gave me the space to move forward in my own time without betraying myself. You are a light in my life and oh, darling, do I love you!*

*And finally, to my family. Particularly my parents Brent and Lidija Phillips; you have been my cheerleaders, supporting me from beginning to end. Thank you for your wisdom, for your care, and for exemplifying a mentality of the hard work and perseverance that I needed to complete this journey. I am who I am because of your love and care.*

## ACKNOWLEDGMENTS

I want to express my gratitude to my committee members for your generosity with your expertise and your patience throughout my journey. A special thanks to my advisor, Dr. Donald Brown; your support and insights have been vital in this endeavor and I am so grateful for your role as my advisor. Thank you to my committee chair, Dr. Laura Barnes, and my committee members, Dr. Afsaneh Doryab, Dr. Michael Porter, and Dr. Rafael Alvarado, for serving on my committee, for your time, your comments, and your wonderful questions that challenged and led to an improved dissertation.

I would like to especially thank my research team that supported this effort and generously gave their time to secure funding, review papers, provide the necessary domain expertise, and develop the initial prototype of the system: Dr. Jennifer Chiu, Dr. Ginger Watson, Dr. Jim Bywater, Dr. Sarah Lilly, and fellow researcher Debajoyti Datta.

Thank you to Zachary Rieman for your sacrifice in countless hours to support a wide range of tasks involved in this undertaking: from system development to dissertation edits.

Finally, thank you to all who helped recruit individuals for the user study, specifically Lidija Phillips and Deborah Rieman for your efforts that went above and beyond to find eligible participants. Your support made the completion of my study possible.

## ABSTRACT

Tech giants have spent millions of dollars developing what appear to be intelligent conversational agents, such as Siri and Alexa[3]. While advancing portions of the text analytics field, these applications often rely on vast amounts of pre-programmed tasks and rules-based dialogue policies, as well as a large amount of domain expert input to achieve the illusion of language understanding. The illusions created by either well-funded or straight-forward, closed-domain, task-oriented, and typically customer-service-centric conversational agents have led to a societal-level misconception of what we are truly capable of within the Conversational Agent domain. In domains with insufficient data and funding, the hopes of developing complex, diverse-purposed conversational agents are often unlikely to be realized due to the lack of labeled data, resources, and codified processes that differ from customer-service-oriented design needs.

In this dissertation, I detail the process of developing a meta-purpose conversational agent, specifically a pedagogical teachable agent. This development is one of few meta-purpose agents in the literature and the first pedagogical teachable agent in the literature that incorporateds state-of-the-art Natural Language Processing (NLP) techniques such as incorporating generative responses and free-form natural language user inputs. Users engage in a teacher role when interacting with our virtual student, the AI-based classroom teaching system (ACTS), who needs assistance with a STEM-related problem. I outline discussion for evaluation needs for non-customer-service agents and the importance of anthropomorphic quality development, such as mimicking the fallibility of understanding more representative of a real-world student. I propose a development framework and provide transparent insight into the development process. Finally, I validate the proposed conversational agent with a novel study involving the assessment of my design. The results of this study contribute to pedagogical conversational agent discussions and the development process for meta-purpose and teachable agents.

# Table of Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

Evidence suggests that deliberate practice will improve teachers' mathematical questioning abilities. The opportunities for such practice are not common in pre-service programs or in-service settings due to various constraints in teacher preparation programs. A computer-based system can provide additional opportunities to rehearse skills and receive feedback for improvement. Such a system can provide feedback by developing a Conversational Agent (CA) that acts as a virtual student. Within the education domain, this would be considered a Teachable Agent (TA) as opposed to a teaching agent or a peer agent because the agent is designed so that it learns some subject matter. There are few such teachable agents, and the few that exist are far behind the current state-of-the-art conversational agent implementations. The current teachable agents tend to have little to no text input capability, instead relying heavily on decision trees and simulation models. Additionally, teachable agents in literature are typically designed around content-based understanding in which students learning a topic may benefit from teaching the material they are attempting to learn. I present the development of an AI-based classroom teaching system (ACTS), novel in that:

- It is the first attempt at incorporating a dialogue foundation within the TA category
- It is a meta-purposed conversational agent: the task presented within the discussion does not directly align with the goal or purpose of the conversational agent.

This paper presents the development of an AI-based classroom teaching system (ACTS) designed to help teachers rehearse mathematical questioning strategies that leverage advances in conversational agent (CA) development. In particular, this paper describes the use of a human expert working with the computer-based system in a supervisor-type role to step in and keep the conversation going when the CA may fail. The system's goal is to simultaneously collect data for conversational agent components while maintaining a coherent conversation and relying on the most up-to-date advances in natural language processing systems. This paper reports on the development and user testing of the ACTS system.

Natural Language Processing (NLP) improvements benefit various domains, including conversational agents. A lack of sufficient data, however, especially structured data, makes it difficult to implement the latest NLP methods. Recent NLP advancements have accounted for some progress by developing readily available pre-trained models that lessen the data needs in developing questions-and-answering conversational agents. However, these developments often require structured data to deploy conversational agents to specific domains. The required domain expert knowledge and time to generate the required data remain sizable, and achieving sufficient data needs can prove challenging. The crux of this research lies in further leveraging unstructured data to help alleviate the data needs to develop domain-specific conversational agents. This paper proposes a design to incorporate unstructured data within the knowledge base component of a conversational agent. The resulting modular conversational agent leverages previously developed NLP models as components of the design. Combining techniques incorporating unstructured text minimizes the effort required to generate new scenarios to implement in varying domains.

Conversational Agents (CA) have seen increasing anthropomorphic features to allow for improved interactions in various fields, with the critical elements of improvement relating to the interpretability of user inputs and improved robustness of generated responses. Within the educational domain, more CAs must be developed and designed for skills practice and assessment rather than exclusively as a tutor or lecturer role-fill. There is a gap in the avail-

ability of assessment and feedback mechanisms available to pre-training teachers as they seek to implement teaching skills. This paper centers on a conversational agent designed to provide additional and readily-available assessment and feedback options for pre-service teachers. It is intended for use as a tool in the pre-service curriculum. A critical element that sets this conversational agent apart is the need to represent imperfect knowledge more anthropomorphically. I propose a design for an imperfect knowledge base structure which I refer to as "adapting knowledge". Adapting knowledge refers to the ability of a conversational agent to progress or regress in responses on a given topic without the need for domain expert-crafted hierarchical structures within the design.

Conversational agents (CA) developed over the past decade have seen innovative artificial intelligence methods introduced into application in customer service fields [57], student education[41, 14], and many additional domains. With advancements in Natural Language Processing (NLP), an intensified effort to incorporate anthropomorphic qualities within CAs has increased. One such quality underrepresented in literature is a representation of imperfect knowledge. Literature shows that TAs are developed to simulate student pre-programmed behavior to certain actions within the education domain. For example, if a teacher chooses the action to assign a group project rather than an individual project, a simulated random model among a roster of assigned students may have a different impact on their grades as some students would excel in this while other simulated students grades will suffer for the same action. The idea of TAs is not novel; however, apart from the high-level simulation at a cause-and-effect decision level, no experience or individual skills-based systems have been developed.

A CA that simulates a student's ability to learn requires an alternative design, distinct from typical CAs designed to respond with the most informative or correct response. The key to this nuance, to develop a CA with intentionally imperfect knowledge, lies in the construction of the knowledge base and the logic within the dialogue management system. An effort to develop this imperfect understanding is discussed, and there are further im-

provements for this element in CA design in the future works section. This dissertation uses several techniques to design a knowledge base that better represents a student's imperfect understanding of a topic. The proposed design incorporates an ability for the CA to transition between levels of understanding of a topic; depending on user input, the CA can improve in its responses, simulating the growth of student understanding. It can also regress in its responses, representing confusion on a topic when the user introduces overwhelming amounts of instruction or unclear knowledge. The value of this anthropomorphic quality is applicable within the education domain and any domain seeking to more robustly simulate adapting human understanding - thereby moving away from the hard-coded knowledge base responses of the past to generate more robust answers in simulations. The paper discusses the proposed design of the CA, highlights techniques intended to simulate the attribute to learn, demonstrates preliminary results, and discusses future research to evaluate this design.

This dissertation focuses on simulating an education-domain scenario to utilize modern natural language processing techniques to support the development of less-rigid rule-based conversational agents even within closed domains. I accomplish this by addressing the following research questions:

**Research Question 2.1**: What gaps are within conversational agent development literature?

**Research Question 2.2**: What gaps are within pedagogical systems literature?

**Research Question 2.3**: What gaps are within pedagogical conversational agents literature?

**Research Question 3.1**: Given the comprehensive evaluation strategies for conversational agents presented in the literature, what metrics are applicable and best suited for teachable agents? What additional considerations should be accounted for in meta-purpose agents?

**Research Question 3.2**: What would a process model be to address the design

and evaluation components of a pedagogical teachable and meta-purpose agent?

**Research Question 4.1**: Can I design a system that fills pedagogical needs for individual skills practice, modernizes conversational agent approaches within teachable agents, and adopts a meta-purpose framework?

**Research Question 4.2**: How can I overcome low-to-no-data scenarios to develop critical components of a conversational agent, such as a knowledge base?

**Research Question 4.3**: Can I develop a framework for conversational agent classification-element-development in low to no data scenarios?

**Research Question 5.1**: With insights from preliminary testing, what changes can I implement to improve the design of a pedagogical teachable agent? Can these improvements be demonstrated to allow for transparency in the development process of a conversational agent?

**Research Question 5.2**: In a no data, minimal time scenario, what is an effective way to deploy a new intent classification component within a conversational agent design?

**Research Question 5.3**: When building a dialogue management system, how can I utilize a generalizable structure to minimize the requirements in developing additional scenarios?

**Research Question 6.1**: Given the difficulty associated with comparing niche conversational agents with each other, can I demonstrate establishing a baseline and a gold standard in the conversational agent development process?

**Research Question 6.2**: What insights can I gain from completing a real-world test of the system?

**Research Question 6.3**: How do the proposed evaluation metrics compare with previously identified conversational agent metrics in literature?

The remainder of this work is presented as follows. In chapter 2, I summarize the literature surrounding the discussion of conversational agents generally and within the education

domain. I motivate the problem by identifying a gap within the literature for this type of conversational agent development and the need for metrics and processes for conversational agent development within the education domain. In chapter 3, I discuss conversational agents' development process and evaluation, and I propose a codified process and metric to utilize. Chapter 4 elaborates on the prototype development of my novel design for a pedagogical teachable agent: the Artificial Intelligence Classroom Teaching System (ACTs). In Chapter 5, I continue to utilize the development process from Chapter 3 as I discuss the development of a dialogue management system when starting with no data. Chapter 6 continues the development process to illustrate a system test in a User Study and incorporates a discussion of evaluation metrics and continued iteration next steps for the development process. I conclude with a summary of contributions, a discussion of limitations, and a proposal for the next steps.

# Chapter 2

# Literature Review

To understand the context of the conversational agents within the literature, I address several topics within three sections. First, a discussion of conversational agents, the state-of-the-art technology used to develop conversational agents, and gaps found in the literature, mainly the advancement of Natural Language Processing (NLP). In the subsequent Section, the focus is centered on the context of conversational agents within the educational domain. I discuss pedagogical conversational agents, state-of-the-art for pedagogical conversational agents, and gaps within the literature. Finally, I conclude with the gaps found at the intersection of conversational agents and pedagogical conversational agents.

The research questions answered within this chapter are as follows:

**Research Question 2.1**: What gaps are within conversational agent development literature?

**Research Question 2.2**: What gaps are within pedagogical systems literature?

**Research Question 2.3**: What gaps are within pedagogical conversational agents literature?

## 2.1 Conversational Agents

### 2.1.1 History and Perception

Conversational agents are applications that perform some level of Natural Language Processing (NLP) to make meaning of user input messages and respond. Depending on implementation and task requirements, conversational agents can provide a low-cost solution and are regarded as one of the most promising areas of artificial intelligence technologies as the capabilities for the underlying methodologies continues to advance.

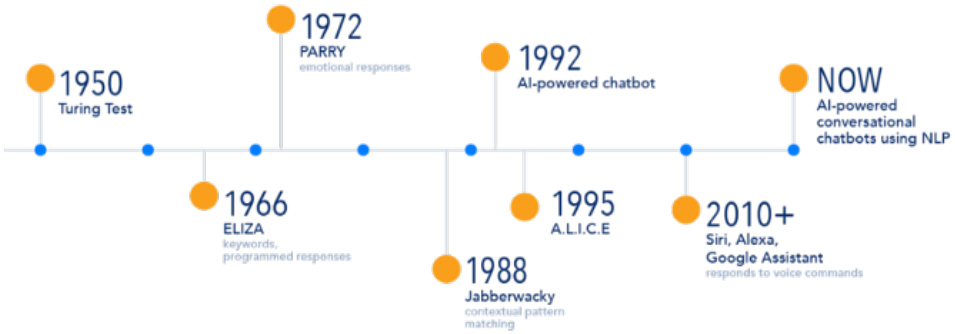



**Fig 2.1.** History of Conversational Agents 1950-2021 [43]

Figure 2.1 displays a timeline of relevant dates in the conversational agent development. In 1950, the Turing test was proposed by Alan M. Turing as a test for intelligence in a computer. Not long after, a prominent, if not the first conversational agent was introduced in 1966: ELIZA [54]. Although the logic that dictated the responses of ELIZA was simplistic, rule-based pattern matching of user inputs, ELIZA successfully convinced some users of being intelligent, to the point where some asked to be left alone with the system to engage in discussion. Even the earliest conversational agents Are pushing the limits of the Turing Test.

Conversational agents have grown in prominence and are considered well adapted in

portions of our present-day society. With tools such as Alexa, Siri, and Cortana saturating the market in Western culture, few in those realms have not yet encountered these tools or chatbots [23]. In interactions with present-day conversational agents, there is often user frustration, leading to users changing their standard speech patterns to engage with systems.

Perhaps most pertinent to that frustration is the disconnect between what users believe conversational agents are capable of understanding compared with their actual ability. This lack of understanding perpetuates the illusion that conversational agents are more capable than they are [26]. This illusion damages the ability to have meaningful conversations while developing conversational agents [26]. The gap discussed is a crucial insight as partnerships across domains less familiar with the technology that supports conversational agent foundational development continues to grow.

### 2.1.2 Natural Language Processing Development

Natural Language Processing is a combined research area of artificial intelligence (AI), computer science, and linguistics. This field focuses on developing the capacity of computers to process and analyze natural language data. Processing is a crucial component to conversational agent development as we move to improve user experience. The foundational elements within NLP were developed over time to become statistical models within the 1990s, such as Bag of Words(BoW) and an advanced variation of BoW, term frequency-inverse document frequency (TF-IDF). This was before what can be referred to as the Deep Learning Era [37]. Figure 2.2 displays the development of NLP from 1949 to the 1990s, capturing key development moments.

Advanced language modeling capability became possible after the statistical model development within NLP. Neural language modeling eventually developed to the point of the shared innovations of large, pre-trained language models in 2018. This is depicted in Figure 2.3.

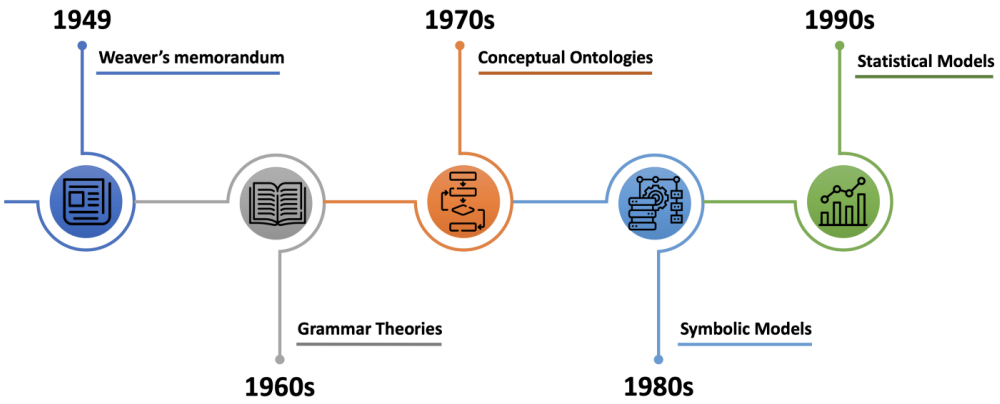



**Fig 2.2.** Natural Language Processing before the Deep Learning Era [37]

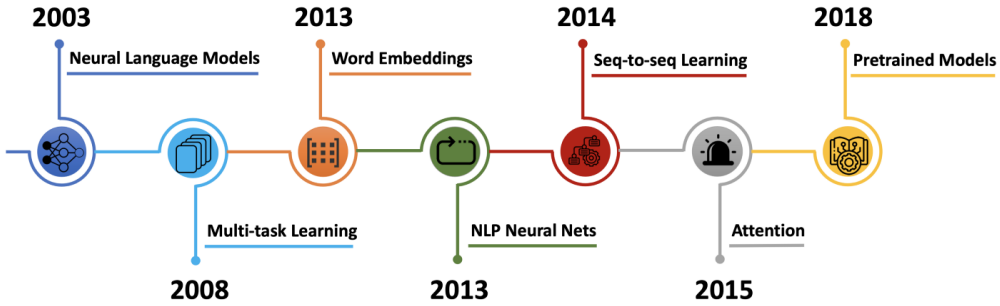



**Fig 2.3.** Natural Language Processing during the Deep Learning Era [38]

### 2.1.3 Conversational Agent Boom

We are at the precipice of emerging NLP technologies. Recent development in NLP technologies has allowed advanced techniques relevant to conversational agent developments to grow and expand.

Two critical components of a successful conversational agent include the ability to understand the user input in a meaningful way and the ability to generate a meaningful response. While there is overlap between the methods used to accomplish each of these tasks, it is worth further specifying the domain boundaries within the natural language processing umbrella. Two additional relevant tasks include automatic speech recognition (ASR) and text-to-speech (TTs), as these are often incorporated in modern conversational agent development. Notably, foundational techniques within natural language processing precede and underpin the methods presented in this figure and are often essential components of the ability to conduct more advanced approaches. Some such techniques included are Bag-of-Words

(BoW), topic modeling, TF-IDF, and word embeddings, to name a few.

Figure 2.4 displays appropriate methods and applications within NLP pertinent to conversational agent development. Figure 2.4 further depicts the categories of NLP techniques that fall into two components: Natural Language Understanding(NLU) and Natural Language Generation (NLG). NLU is computer reading comprehension; within this component are NLP techniques purposed to interpret text and speech to understand the true meaning. NLG is computer generation, or writing text, and includes techniques in NLP that can produce a human language text response based on input.

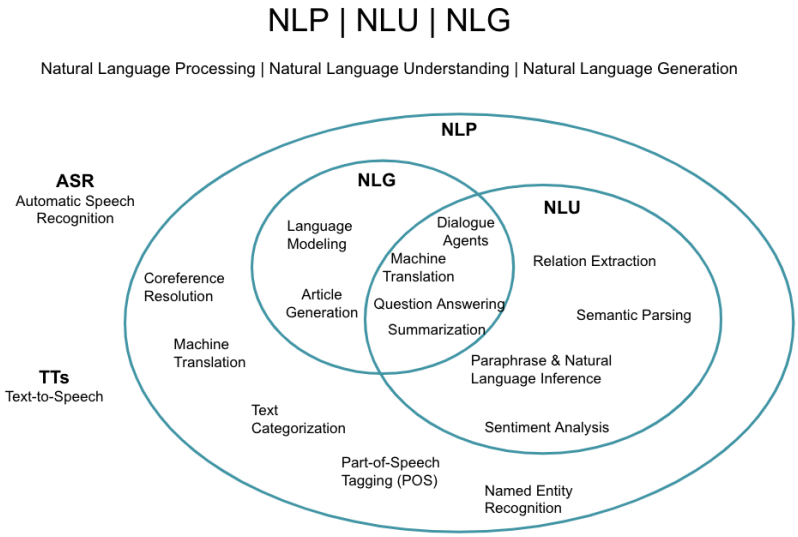



**Fig 2.4.** Natural Language Processing Subfields relevant to Conversational Agents: Natural Language Generation, Natural Language Understanding, Automatic Speech Recognition, and Text-to-Speech

With today's emerging NLP capability, we have moved from simple chatbots that accomplish tasks to true conversational agents capable of interpreting more robust input language from users. There is still a gap in the ability to understand user context, and development in the NLP field must continue if we are to one day understand more naturally colloquial user language.

User experience with conversational agents has also developed over time, moving from

click navigation choices to raw text input. Many interactions still need to be structured, and NLU's capability to understand user input and parse into intents and entities has allowed for keyword recognition and structured phrase matching.

Figure 2.5 shows the user experience vs. the technology development and where the state-of-the-art in current conversational agent capabilities is.

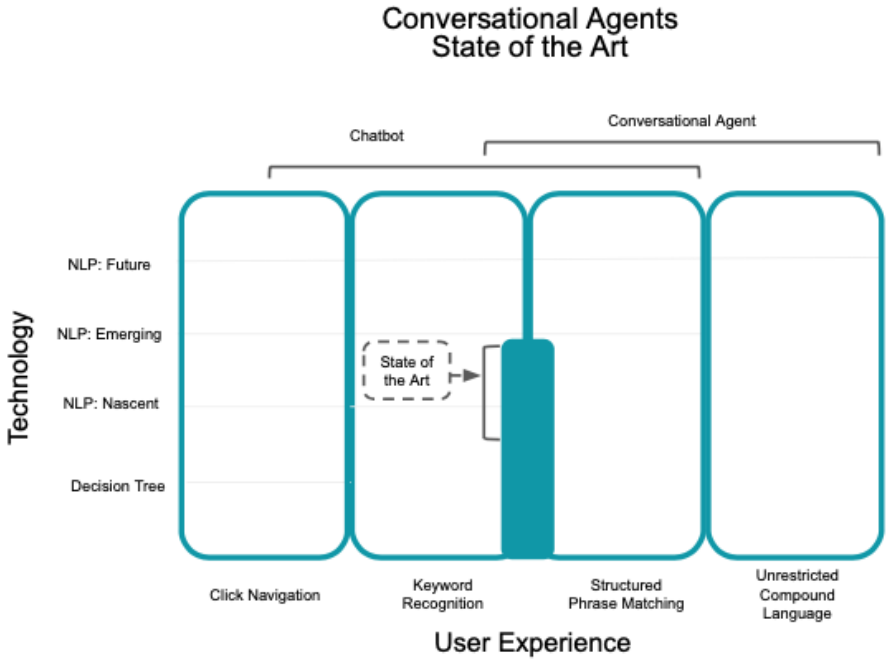



**Fig 2.5.** State of the Art User Experience vs. Technology of Conversational Agent Development

### 2.1.4 Conversational Agent Structure and Framework

Given the advancement of methods within natural language processing, the ability to move beyond a rules-based framework in conversational agent design has become feasible. Figure 2.6 provides a depiction of capabilities when designing conversational agents. Previously, only closed-domain conversational agents were feasible, as technology for machine learning had not yet matured. A closed domain conversational agent is narrowly focused on specific tasks and topics and cannot process inputs outside of its specialized domain. Because open-domain conversations are so diverse and unpredictable, it is infeasible for a rules-based dialogue system to function in such conversations - the decision tree and conditions would

be infinite. However, machine learning has opened the door to the next level of technology, retrieval-based systems. For example, one may use semantic matching on user inputs to find the most semantically similar piece of information from a database. This utilizes techniques found within information retrieval in order to find topically relevant responses. However, current technology has advanced beyond retrieval models to systems described as generative. Generative responses are when the conversational agent can take input and generate relevant output without basing the output on a version of the input text. One step in between these two types is a retrieval-generative model where a semantically matched piece of information along with user input may be used as inputs into a transformer model that may then output a wholly new text based on those two inputs. This text was not explicitly dictated and was crafted based on the transformer model.

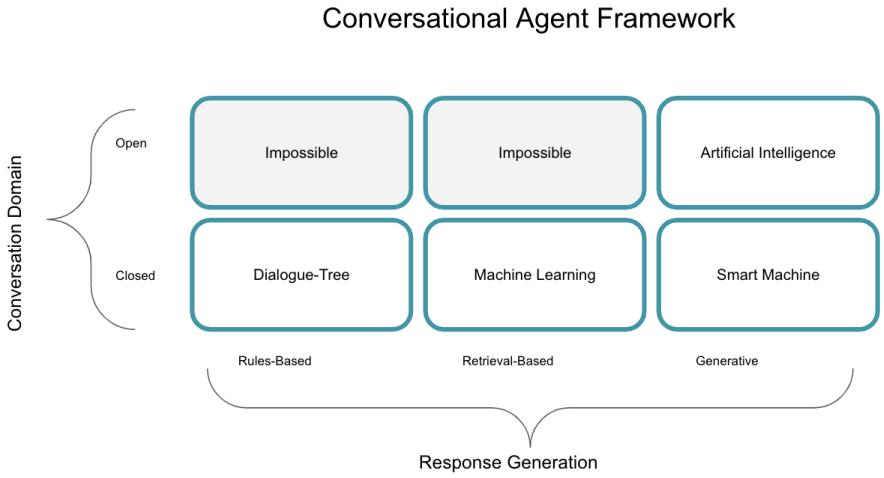



**Fig 2.6.** Frameworks for Conversational Agents

The key components of modern conversational agent architectures include the following elements:

1. **Intents** : what the user intends, or what their true goals and purposes are in submitting the input text
2. **Entities** : the elements or slots that capture pertinent data to the intent
3. **Dialogue Management** : the architecture and flow of how the user input is transformed into meaning. At a minimum, this incorporates the logic for intent and entities

and captures how a computer response is generated based on the user input.

4. **Database** : a knowledge base of data that is used in referencing logic by the dialogue management component of the system

An example of a basic structure of a conversational agent is depicted in Figure 2.7. This shows a simple example of a conversational agent that can be used to book a flight. The user puts in the request, and the conversational agent identifies the user's intent as booking a flight. The pertinent entities in the example include the departing airport, arrival airport, date, time of day for the flight, and the price of the flight. These are captured within the NLU component of the conversational agent. The information of an identified option for flight is then provided to the user for consideration.

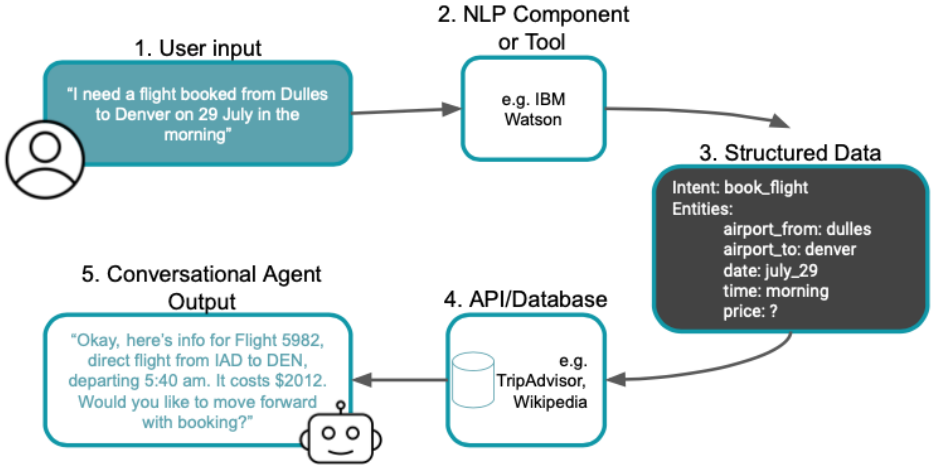



**Fig 2.7.** Basic Architectural Components of Conversational Agents

### 2.1.5 Development Requirements

Resources to develop any conversational agents are typically high. Rules-based systems require expertise in understanding patterns of input as well as large amounts of time developing a robust dialogue tree and maintaining it. Alternatively, generative and retrieval-based systems require trained data and knowledge stores to develop a conversational agent that can engage in more meaningful dialogue [52].

Recent technological advancements such as introducing probabilistic models to conversational agent architectures have expanded the utility of conversational agent developmental

efforts that may have fewer data and resources [51]. Additional advancements in methods that allow for lower resource strains include weak supervision and transfer learning, which allow for a lessened resource load in developing models for a conversational agent. These can be found in use more and more regularly, but the most known conversational agents typically achieve better abilities in dialogue due to the resources and more manual methods that underpin their designs.

Examining several recent surveys of conversational agents also reveals that most designs in the literature suffer from excessive specificity of design and a lack of reproducibility [45, 42, 58, 32]. Even with code provided, the skills necessary to successfully employ conversational agent code can limit the ability of non-AI-centric domains to utilize resources available to them. One survey found that a lack of familiarity with programming skills was a factor [49, 51]. Together these limitations reveal an area where the dialogue of conversational agents can be opened to a wider variety of domains and skill sets by improving the transparency of systems developed and improving the accessibility of these systems.

An additional consideration for developmental conversational agent is **Meta-purpose-agents**. A Meta-purpose agent is a conversational agent with a primary purpose other than the task or dialogue content within a user-engagement session. Many customer service conversational agents have a task orientation to support the user. They are developed and designed to fulfill various tasks such as booking flights, providing information, completing shopping, supporting complaints and issues, and rerouting users to the appropriate human. The content of the dialogue in the sessions directly corresponds to the purpose the conversational agent is developed and designed for. A meta-purpose agent could be a system designed to represent a virtual student who needs help learning a concept. The dialogue in this example would be centered on teaching the student the learning concept. The purpose of the conversational agent, however, may be for the user to gain experience, feedback, and insights that would improve their pedagogical questioning skills. The purpose is not based on the content of the dialogue but rather the way the user engages the conversational agent

and the exposure to an additional skills practice opportunity. Evaluation of a task-oriented conversational agent tends to be limited to efficiency and few anthropomorphic qualities that represent the most desired qualities in a customer service interaction, such as friendliness and clarity. In the meta-purpose agent example discussed, efficiency is not the purpose but rather a more anthropomorphic evaluation to determine the ability of the conversational agent to mimic a student is desired. Students are not necessarily efficient; they are not necessarily friendly; the desire in this interaction may be exposure to a student's varied understanding and personalities that a pre-service or in-service teacher may encounter.

As the technology advances and state-of-the-art techniques allow for greater mimicry of human capabilities, there will be an increasing number of meta-purpose agents and an increasing need for design and evaluation considerations of anthropomorphic-oriented systems to include sub-optimal logic and imperfect knowledge.

### 2.1.6 Gaps in the Literature

The following gaps within the literature are identified within general conversational agent development efforts:

> **Conversational Agent Gap 1**: *Processes for development in low to no data scenarios.* Most publications and discussions of conversational agent development assume high-resource and data capabilities.
>
> **Conversational Agent Gap 2**: *Accessibility of state-of-the-art development methods.* Requirements to understand, design, and build a true conversational agent incorporating natural language processing emerging techniques are limited by highly technical understanding as well as programming skills. Additionally, literature often leaves out critical steps in the development process, access to data, and reproducibility of systems is nearly impossible due to black-box publications. There is a need for higher transparency and simplification of the implementations proposed.
>
> **Conversational Agent Gap 3**: *Diversification of conversational agent purpose:*

*Meta-purpose Agents.* The majority of literature discusses customer service purposed conversational agents. An even higher percentage of literature discusses task-oriented chatbots. Little discussion of conversational agents developed as a tool to practice skills external to the content of discussion with a system that adds additional layered considerations in design and development.

## 2.2 Educational Domain Conversational Agents

### 2.2.1 Types of Pedagogical Conversational Agents

Conversational agents within the education domain can be separated into three main categories: teaching agents, peer agents, and teachable agents [11]. Teaching agents provide knowledge to the user and help instruct the user by providing feedback and information throughout the interaction. A peer agent is one where the user and the agent learn a topic together, and through eliciting questions, a user may better consider the topic and internalize the information. A teachable agent is one where the user instructs the agent on how to accomplish a task and provides the information to the agent. Of note is that within the literature, teachable agents are content-based, where users interact with a system in order to learn the topics they themselves will practice teaching during the interaction.

Of these categories, there is compelling evidence for the impact of teachable agents in supporting learning and evidence that this is the most underrepresented category of conversational agents in literature [11]. Within the three years since publishing, little has changed. A literature review of pedagogical conversational agents published in July 2022 found the majority, 55.5%, of pedagogical conversational agents to fall within the teaching category, 36.11% to fall within the peer agent category, and only 5.4% within the teachable agent category [30]. Figure 2.8 shows several examples of each type of pedagogical conversational agent, including all of the teachable agent dialogue agents our team identified in the literature.

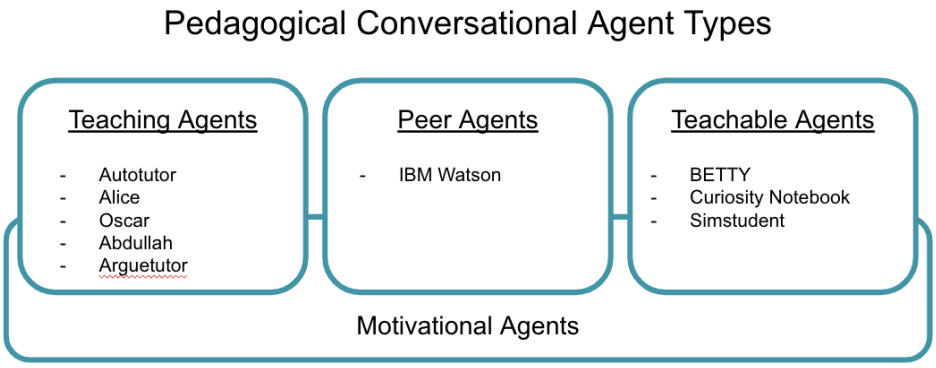



**Fig 2.8.** Pedagogical Conversational Agent Types

### 2.2.2 Teachable Agents

The teachable agents identified include SimStudent [39], Curiosity Notebook [33], and BETTY [9]. These examples are all purposed for content-based learning where users engage the conversational agent to better learn the content they are teaching. Additionally, user input for these systems is limited to a minimal open dialogue where users typically train and model within a system through clickable options and decisions. The only text options are preset and non-generative.

This gap within literature is further represented in Figure 2.9 where I elaborate further on Figure 2.5 and have inserted a depiction of teachable agents found in the literature. Here I depict the state-of-the-art where the user experience is reflected on the x-axis, and the technology capability is reflected on the y-axis. An unrestricted language conversational agent would allow users to engage as if with a human and be understood and responded to as if the agent was human. The Nascent NLP technology represents foundational techniques in NLP such as TF-IDF, and "NLP:Emerging" methods include techniques such as co-referencing and language modeling. The technology readily available today allows conversational agents to move beyond click navigation to improve user experience; however, the teachable agents found within the literature have all been decision-tree and click-navigation-based.

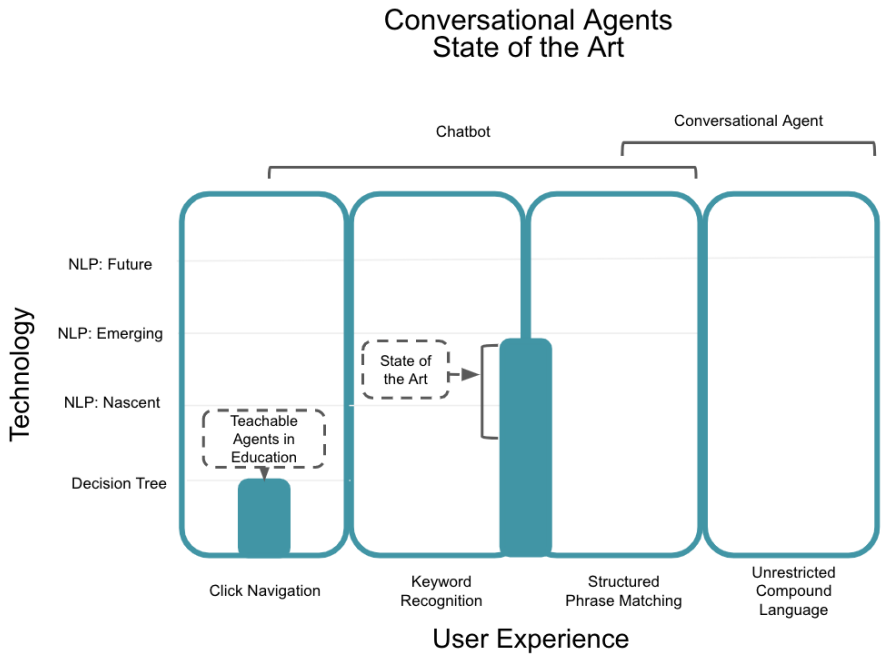



**Fig 2.9.** Pedagogical Conversational Agent State-of-the-Art

### 2.2.3 Pedagogical Skills Practice

Within the education domain, another gap is found in systems purposed to support skills learning for educational skills practice at the individual level [7]. Figure 2.10, based on research done by [7], identifies the gap between digital skills practice opportunities and at an individual skills level. While digital systems exist, these systems utilize human-avatar interactions [13] and higher-level classroom management [39].

Teacher questioning skills have been demonstrated to be a critical component of promoting successful mathematical discussions [44, 2]. With a gap in digital technologies available at an individualized level, this highlights an area for further pedagogical conversational agent development allowing teachers to train in this vital skill.

### 2.2.4 Gaps in the Literature

> **Pedagogical Gap 1**: *Modernization of Teachable Agents.* Zero (0) teachable agents found in literature incorporate emerging NLP technologies or improved maturity in user experience.

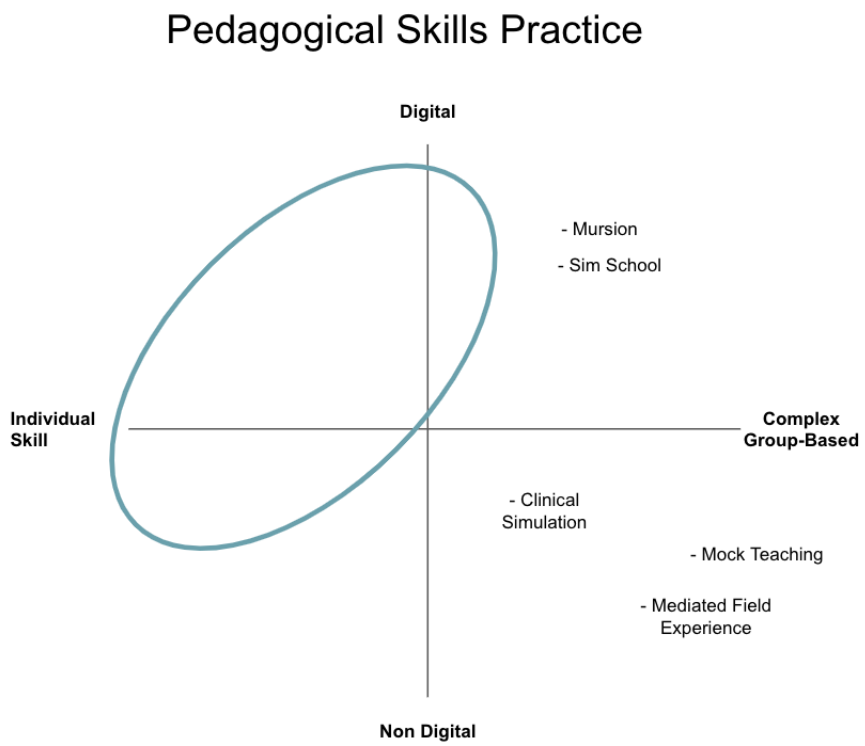



**Fig 2.10.** Pedagogical Skills Practice Framing Gaps Identified in Literature [7]

**Pedagogical Gap 2**: *Digital System Development for Individual Skills Practice.* No digital systems exist that address increasing individual skills practice for pre-service and in-service teachers.

## 2.3 The Intersection of Pedagogical Needs and Conversational Agent Development

Conversational agent literature is saturated with development efforts centered on customer service goals. This is equally as true for the many publications outlining evaluation strategies for conversational agents. While in Section 2.1.6 discusses the need for diverse conversational agent purposes, it does not incorporate the evaluation strategies specific to the niche at the intersection of Pedagogical purposes and conversational agents. There is a clear need for comprehensive evaluation strategies that are valid for both meta-purpose and pedagogical

conversational agents.

Additional literature reviews within the pedagogical conversational agent subdomain reveal gaps that highlight areas where further development is needed. Some of these gaps are because of the heavy focus on task-based efficiency conversational agents that are better utilized in industry settings and do not translate well within the education domain. The findings of this review highlight these gaps: [22]

> **Niche Gap 1**: *Generalizable Design Knowledge* Most pedagogical agents in the literature fail to discuss or provide in-depth transferable insights.
>
> **Niche Gap 2**: *Comprehensive Evaluation Strategies.* There is a lack of comprehensive evaluations covering multiple aspects such as learning success, technology acceptance, software quality, algorithmic quality, and suitability of application scenarios.
>
> **Niche Gap 3**: *Conversational Agent-level Development Process Model* There is a lack of process models addressing the design and evaluation of pedagogical conversational agent systems.

## 2.4 Conclusion

To summarize, I have identified three gaps within the general conversational agent literature, two gaps in pedagogical system development, and three gaps within the niche intersection of the educational domain and conversational agent development. I have answered the research questions posed at the beginning of this chapter.

To this end, the remaining portions of this dissertation are devoted to developing ways to address portions of the gaps mentioned. I additionally aim to implement two best practices within the machine learning community in addressing these gaps: avoiding black box development and utilizing as simple and precise language to convey meaning and purpose as to be interpretable.

# Chapter 3

# Conversational Agent Evaluation and Development Process

Evaluation has long been a topic of discussion for conversational agents. The Turing test attempted to establish a metric for intelligence decades ago. Yet, shortly after ELIZA debuted in 1966 [54], the debate of whether there were better ways to measure conversational agents and intelligence arose. The literature is full of discussion on conversational agent metrics, yet no undisputed metric exists. Additionally, metrics and processes discussing conversational agent development are often centered on task-oriented and industry-purposed chatbots, thereby not meeting the needs or relevance of pedagogical agents. In this chapter, I discuss metrics proposed in the literature and the applicability to a meta-purpose pedagogical teachable system. With the evaluation component in place, I take the subsequent steps to codify the process of conversational agent development. This directly answers the gap in the pedagogical conversational agent domain [30] that describes a need for processes to be articulated and established within this domain. The research questions that are responded to within this chapter are as follows:

**Research Question 3.1**: Given the comprehensive evaluation strategies for conversational agents presented in the literature, what metrics are applicable

and best suited for teachable agents? What additional considerations should be accounted for in meta-purpose agents? *This addresses gaps in the literature for Niche Gap 2 in Section 2.3*

**Research Question 3.2**: What would a process model be to address the design and evaluation components of a pedagogical teachable and meta-purpose agent? *This addresses gaps in the literature for Niche Gap 3 in Section 2.3*

In this Chapter, I discuss evaluation strategies of conversational agents and propose a method best suited for niche conversational agents. I continue with a discussion of the development of conversational agents and offer a codified process that is then implemented in Chapters 4-6.

## 3.1 Evaluation

"What you measure affects what you do. If you don't measure the right thing, you don't do the right thing." - Joseph Stiglitz

### 3.1.1 Evaluation Metrics in Literature

Measurement impacts development and is a critical consideration in the development process. There is a pursuit in literature for a standardized quantitative metric for conversational agents. As many papers have addressed evaluation metrics, few succeed, and there is a greater acknowledgment of the lack of practicals for such a measure. In the majority of recent dialogue system evaluations, human evaluation is used [18]. The literature reviews on conversational agent evaluation metrics often propose a metric of use; however, I argue that what some have concluded as a standard metric does not apply to all cases. I propose that developers must take unique intention in developing evaluation metrics for conversational agents. This is not a novel claim, and many evaluation-centric papers discuss the need for niche evaluation metrics depending on conversational agent design and purpose. In this section, I discuss two relevant metrics that appear to be more appropriate than most frameworks: naturalness evaluation of conversational agent dialogue systems and a simplified

**Table 3.1.** Proposed Naturalness Evaluation for Dialogue Systems [24]

| **Metric** | **Type** | **Data Collection Method** |
|---|---|---|
| Total Elapsed Time | Efficiency | Quantitative Analysis |
| Total number of user/system turns | Efficiency | Quantitative Analysis |
| Total number of system turns | Efficiency | Quantitative Analysis |
| Total number of turns per task | Efficiency | Quantitative Analysis |
| Total elapsed time per turn | Efficiency | Quantitative Analysis |
| Number of re-prompts | Qualitative | Quantitative Analysis |
| Number of user barge-ins | Qualitative | Quantitative Analysis |
| Number of inappropriate system responses | Qualitative | Quantitative Analysis |
| Concept Accuracy | Qualitative | Quantitative Analysis |
| Turn correction ratio | Qualitative | Quantitative Analysis |
| Ease of usage | Qualitative | Questionnaire |
| Clarity | Qualitative | Questionnaire |
| Naturalness | Qualitative | Questionnaire |
| Friendliness | Qualitative | Questionnaire |
| Robustness regarding misunderstandings | Qualitative | Questionnaire |
| Willingness to use system again | Qualitative | Questionnaire |

evaluation metric for assessment of conversational agents.

In the first example, the proposed metrics are an attempt to provide a method to evaluate the naturalness of a dialogue system [25]. The proposed metrics are listed in Table 3.1. These metrics include a mixed-methods approach detailing both quantitative and qualitative analysis.

These measures address multiple facets of importance with dialogue systems such as composure as well as non-manual metric collection such as total elapsed time.

In a non-task-oriented conversational agent or conversational agents that are designed with meta-purposes, sub-optimal dialogue is not only desired but an essential component. By suboptimal, I mean that the purpose of the conversation is not to accomplish a task or retrieve information as quickly and efficiently as possible; perhaps it is to extend a conversation out further or tell a joke and communicate colloquially.

A conversational agent may be developed as a virtual student for a pedagogical teachable agent purpose. In this case, the metrics presented in Table 3.1 do not all apply, although some may. For example, a student may or may not be clear in their responses depending on

**Table 3.2.** Proposed Simplified Evaluation Metrics [18]

| **Dimension** | **Definition** |
|---|---|
| Grammaticality | Responses are free of grammatical and semantic errors |
| Relevance | Responses are on-topic with the immediate dialogue history |
| Informativeness | Responses produce unique and non-generic information that is specific to the dialogue content |
| Emotional Understanding | Responses indicate an understanding of the user's current emotional state and provide an appropriate emotional reaction based on the current dialogue context |
| Engagingness | Responses are engaging to user and fulfill the particular conversational goals implied by the user |
| Consistency | Responses do not produce information that contradicts other information known about the system |
| Proactivity | Responses actively and appropriately move the conversation along different topics |
| Quality | The overall quality of and satisfaction with the dialogue |

their skill in the topic discussed. They may or may not be friendly, and they may or may not be accurate. Additionally, is efficiency a desired goal for a skills-based pedagogical agent? Is avoiding re-prompts a desired outcome for a system where a human student may likely cause re-prompts of a teacher? These measures appear to apply well in a conversational agent designed for customer service; however, when consider a conversational agent with the anthropomorphic qualities of imperfect understanding and speech, these measures do not all apply. In actuality, the majority represent an opposite metric or goal.

A second approach identified evaluation metrics provided throughout the literature for conversational agents [18]. In this approach, they surveyed the literature and identified redundancy of common metrics and measures. In response, they simplified the metrics and proposed a structure that addressed the majority of metrics found in literature in a meaningful way. These metrics are found in Table 3.2.

These evaluation metrics also provide an interesting consideration; however, upon closer inspection, they again do not appear to align with the purpose of a pedagogical agent intended to represent a virtual student's imperfect understanding. This is true not only for this survey but across conversational agent evaluation discussions in the literature. Given

the lack of applicability despite well-formulated surveys in the literature, I opt to combine a mix of standard practice as well as more specific questions and propose this as a portion of the metric for consideration.

### 3.1.2 Proposed Evaluation Metrics

The evaluation metrics in the literature lack applicability for diverse conversational agent purposes external to a customer-service-centric domain. In developing a conversational agent, I propose three primary components that developers should consider in the design and evaluation. These considerations allow developers to isolate elements for assessing the conversational agent to the greatest extent possible, clearly providing feedback on the impact of each of the conversational agent's essential components on user satisfaction with the system.

The three metrics are displayed in Fig 3.1. First is the User Interface, meaning the components of a system and the way a user engages it. This could range from speech-to-text capability, click and flow process, to the interface color choices or the ability to use the "Enter" button on a keyboard when submitting text. These components impact the user experience, although they are often related to the interface design rather than the Natural Language Processing components found within the logic of the dialogue management portion of a conversational agent architecture. The second category is the output from the computer, which incorporates the components that make meaning of the user input and the dialogue management process to produce an output by the system. Finally, the inherent bias captures users' internal bias despite a potentially perfect system. Even if the language of the conversational agent is perfectly representative of the intended goal, such as an example virtual student, a user may have a bias that influences their ratings despite a perfectly crafted system.

The aspects that developers should consider for a pedagogical teachable agent that is purposed for pre-service skills practice include that it is unique in that the system is not intended to respond correctly always. Not all real students have infallible logic, and there is a diversity in ways of thinking as well as the ability to retain content. Measuring correct-

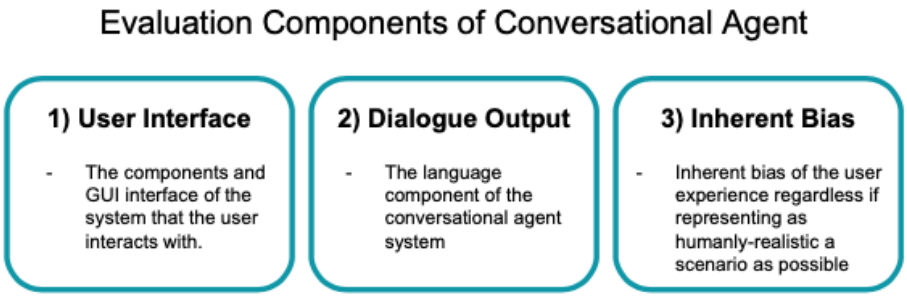



**Fig 3.1.** Evaluation Components to Consider in Conversational Agent Design and Testing

ness, sensibility, and specificity will not represent the desired outcome for a virtual student conversational agent.

Sensibility and specificity are two common metrics in natural language dialogue systems (these fall within the simplified metrics displayed in Table 3.2. I propose maintaining these to examine their validity with real-world data. I also suggest incorporating elements of common standards within the interface metrics, such as if the user would like to interact with the conversational agent again and direct interface questions. Finally, I focus on realism as the realism of the scenario and the student responses are ultimately what I propose is that we desire the system to grow in. The issue with this question is the subjectivity: unique participants and users will have different perceptions of what realism means to them. This subjectivity is a limitation of focusing on these metrics; however, this is only one portion of the evaluation proposal. These metrics are displayed in Table 3.3.

### 3.1.3 Deconflating Evaluation Components

In addition to capturing metrics relevant to the system, structuring the test and evaluation component are key. One way to rephrase this is: with such varied conversational agent purposes and impracticability of comparison despite common metrics, what is a way to capture a "gold standard" for the system in question?

One way to establish what "good" is or a goal for a system is through a user study with a design of experiments. This approach allows for the essential de-conflation of variables such as user perceptions of the interface vs. the language component of the conversational

**Table 3.3.** Proposed Qualitative Survey Questions

| Category | Survey Question |
|---|---|
| Specific | The student responses were appropriately specific. They demonstrated an understanding of my inputs |
| Sensible | The student responses were sensible |
| | The student responses were clear |
| | The student responses were logical |
| | The student responses were consistent |
| | The student responses were normative |
| | The student responses were confusing |
| | The student responses were illogical |
| | The student responses were contradictory |
| Student Realism | Dialogue with virtual student was representative of actual student interactions |
| | The virtual student's understanding of the problem was representative of an actual student |
| | I am satisfied with what the virtual student understood at the end of our interaction |
| | Responses from the virtual student showed increased understanding or learning over time |
| | The virtual student's responses (dialogue) were representative of actual students |
| | The virtual student understood my questions |
| | The virtual student's responses represented student understanding |
| | Timing of student dialogue was realistic |
| Scenario Realism | The conversation flowed in a way that I would expect |
| | I was able to ask realistic questions |
| | The system allowed for realistic exchange between the student and me |
| Interface | Visualization of the mathematics problem supported dialogue |
| | Visualization of the mathematics problem aided dialogue during teaching |
| Desire to Interact Again | I would want to chat with this student again |

agent. The first step in pursuing this is to identify the "gold standard" for a system. For a pedagogical teachable agent, the "gold standard" may be represented by a middle school student. Whatever the system is designed for, the "gold standard" should represent what the near-perfect version of the system could be.

The design of experiments (DoE) should be constructed to allow some version of the existing system to be tested as one configuration and a second configuration a human as near as possible to the desired representative for which the system was created. In the example of a pedagogical teachable agent, this can be represented as a human student, or if that is not possible or impractical, then a human with set guidelines for how to behave as a student might behave in a similar situation. This blind experiment will allow for an established baseline of the system in question as well as an established "gold standard". While some evaluation metrics request input on the system interface, comparing the "gold standard" over time captures some movement that a user may be unaware of themselves. Humans are perceptible to presentation, and the way a system interface is designed and how it allows a user to interact with it, the capabilities built-in, will impact the user's perception even if it is not conscious to the user explicitly.

If conducted appropriately according to a design of experiments approach, the results can help developers identify the true gap in the system's performance. An implementation of this evaluation is conducted and further discussed in Chapter 6.

## 3.2 Codifying Development Process

Within research and the literature, conversational agent development is often discussed finitely- there is one single development process, and it is carried to completion.

### 3.2.1 Industry Standard

This is known as a "Waterfall" framework in software development and is shown in Figure 3.2. This strategy is often suited for a simple and unchanging environment where developers can specify all the system needs at the start of the design process. This is not the case for

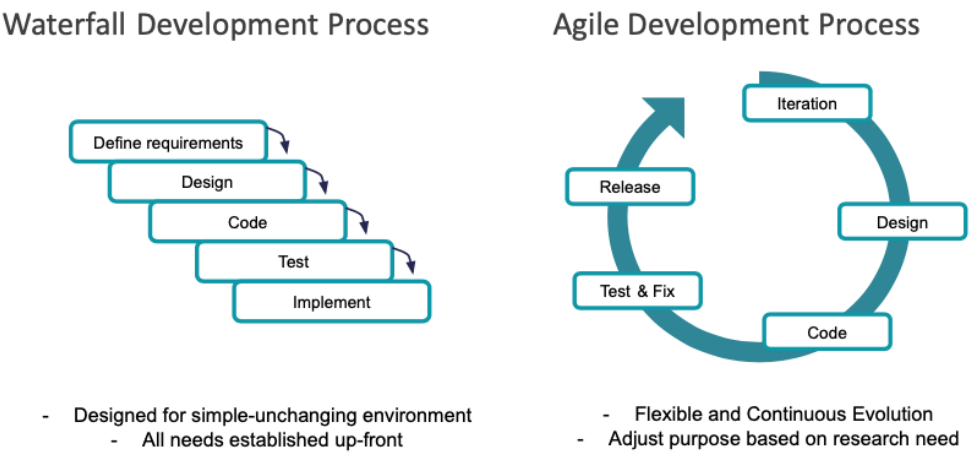



**Fig 3.2.** Software Development Frameworks

research development projects, nor for complex systems that desire to incorporate state-of-the-art techniques. For these systems, an Agile framework is better suited, also displayed in Fig 3.2. In an Agile development process, developers iterate over the design to develop improvements and continually code, test, fix, release, and then redesign. This iterative process allows for the flexible and continuous evolution of a system. It allows for continued integration of newer methods or components within the design needs of the system are unclear, and it is better suited for scenarios where design needs before development. In a research environment, this method is a more logical choice although it is not the approach often pursued, perhaps because researchers often move on to alternative projects before they can pursue iterations on a system.

### 3.2.2 Proposed Process

The specific development process I submit is outlined in Figure 3.3. While drafting the basic idea for a conversational agent, identify the key components - in our case, this was the interactive image designed to facilitate a teacher-student experience around a mathematical question on scale factor. At this point, the technical groundwork for finding that gold standard will also need to be set, with different "modes" where the facilitator may be involved in a different role. Once the first "dummy" interface has been deployed, prototype conversations can be captured, and an iterative improvement cycle can begin.

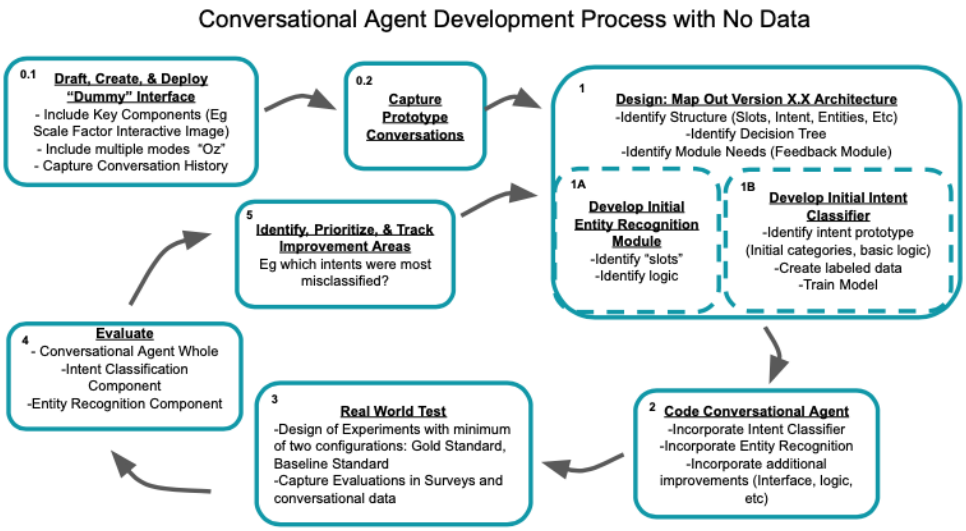



**Fig 3.3.** Process for Research-Based Conversational Agent Development with No Initial Data

In this cycle, I examine the results of the initial version - the "dummy" interface, and the results of the last improved system iteration. At this point, I map out the various new modules required for the system. My system, for instance, required an entity recognition module so as to have at least some baseline awareness of the key components around which the interaction is taking place. The system also needed an intent classifier, a trained model to make sense of user input. At this step, it is also possible to reevaluate what other modules and functions may be required.

From there, it is a matter of developing and building the previously identified modules and then applying them in a real-world test. I propose a gold standard (GS), a version where the pre-service educators interact only with the facilitator impersonating a student, not the conversational agent. Aditionally I propose a baseline standard version(BS) which is a version where the conversational agent interacts as much as possible, and the facilitator only steps in if the agent is unable to respond. By comparing data from interactions between GS and BS configurations, researchers can find specific target areas for improvement. Evaluation of the GS and BS data will reveal areas for improvements like consistently misclassified user input, scenario unrealism that affected the overall score, and more. From there, it is once again a question of mapping out what modules need to be added or improved (possibly trained on

the new real-world data) and what aspects of the interface or scenario need to be improved.

This development process allows for iterative development of a conversational agent that grows more realistic with each iteration. However, the main component of this cycle is the evaluation (via comparison with the Gold Standard) to determine what areas require work. As such, the question of how to evaluate conversational agent success is crucial to the successful development of a pedagogical teachable agent.

## 3.3 Conclusion

In this chapter, I discussed the lack of relevant evaluation metrics for conversational agents with diverse purposes. Specifically focusing on an example with pedagogical teachable agents, I discussed the key elements to consider in the evaluation and how to establish a "gold standard" in evaluating the system and de-conflate inherent bias, user interface, and perceptions of the dialogue output in a system. Finally, I proposed a modified implementation of the Agile framework within the context of conversational agent development and provided details for the implementation steps of this process. Furthermore, these steps are implemented in Chapters 4-6 as an example.

I addressed the following research questions in this chapter:

**Research Question 3.1**: *Given the comprehensive evaluation strategies for conversational agents presented in the literature, what metrics are applicable and best suited for teachable agents? What additional considerations should be accounted for in meta-purpose agents?*

I propose a modified evaluation metric that combines conversational agent evaluation metrics in literature established natural language metrics and software design metrics to address Meta-purpose Agent considerations as well as pedagogical-specific concerns in anthropomorphic virtual student design.

**Research Question 3.2**: *What would a process model be to address the design*

*and evaluation components of a pedagogical teachable and meta-purpose agent?* I leverage a software industry development process, the Agile model, to address pedagogical conversational agent design research needs. I propose a novel development process based on modifying the agile framework incorporating in-depth insight specific to conversational agent development and meta-purpose agent designs. Our model optimizes the ability to integrate continuously emerging technologies which is essential in the conversational agent field of study.

# Chapter 4

# Artificial Intelligence Classroom Teaching System (ACTS) Prototype

## 4.1 Motivation

In this chapter, I demonstrate the development of a prototype conversational agent: an AI-based classroom teaching system (ACTS). The ACTS system is a pedagogical skills-centric teachable agent with the purpose of providing pre-service teachers an opportunity to practice their math questioning skills. Several considerations in our development process include the purpose of the system as a skills-based teachable agent, the role and type of the conversational agent we develop, and a low-to-no-data initial environment. As discussed in Section 2.2.3, significantly few conversational agents used in education use recent advances in machine learning and deep learning, instead relying on simple decision trees [52]. The lack of implementation of current technology reinforces the need for more research and development on artificial intelligence-based methods to support content-specific conversations. I discuss key elements of our development, including the development of a feedback component to address the skills-based purpose of the conversational agent (utilizing artificial intelligence-based methods), the development of a knowledge base component that leverages unstructured text, the process of gathering and labeling data efficiently, and the results of our improved methods of labeling. Finally, I walk through an example of our prototype demonstration.

While the other chapters in this dissertation utilize the first person, in this chapter, I use the term "we", as the prototype was developed with the support of a team and through a joint effort, and each step from pre-processing transcripts to conducting label data collection was approached jointly. Each component discussed in the context of this dissertation was directly worked on by me unless otherwise noted (Such as the coding of the Oz component or the development of the interface code in Django). I did not do the following work without support from fellow team members, and much of this chapter is captured in the published work cited here [15].

I address the following research questions in this chapter:

**Research Question 4.1**: Can we design a system that fills pedagogical needs for individual skills practice, modernizes conversational agent approaches within teachable agents, and adopts a meta-purpose framework? *This addresses gaps in the literature for Conversational Agent Gap 1 in Section 2.1.6, Pedagogical Gap 1 in Section 2.2.4, and Pedagogical Gap 2 in Section2.2.4*

**Research Question 4.2**: How can we overcome low-to-no-data and develop critical components of a conversational agent, such as a knowledge base? *This addresses gaps in the literature for Conversational Agent Gap 1 in Section 2.1.6*

**Research Question 4.3**: Can we develop a framework for conversational agent classification-element-development in low to no data scenarios? *This addresses gaps in the literature for Conversational Agent Gap 1 in Section 2.1.6, Conversational Agent Gap 2 in Section 2.1.6, and Niche Gap 1 in Section2.3*

I desire to support and develop a system that contributes to the literature and provides a baseline to continue further development.

A critical challenge of developing dialogue systems is the need for large data sets for many of the components of the dialogue system, such as intent classification, entity recognition

or slot filling for dialogue state tracking, and the dialogue management system architecture. Rarely do large datasets relevant to a specialty or non-customer-service-centric domain exist. In the education domain, data collection is a significant challenge as domain expert annotations within publicly available datasets are not common, likely not relevant to the specific need if found, and resource intensive to collect. In developing the initial classification component of the ACTS, we collect a. relatively small amount of annotated data. We use approximately two thousand sentences labeled with one of four feedback classifications using the modified IQA classes developed in coordination with education domain experts. The overall implementation of the classification component in the ACTS and the initial deployment of the architecture results in a system that does not necessarily perform better than a conversational agent that has had intensive development with extensive data and resources in its construction. The novelty of this system is that the dialogue management system was developed with minimal resources in a no-to-low data scenario. The initial ACTS deployment provides a starting point to further build to more advanced system versions. It has done so with a minimal amount of domain-expert annotated data.

## 4.2 Proposed Architecture

In our initial implementation and development effort of the ACTS system, we proposed several iterations of architectures. The final attempted architecture we worked to develop is shown in Figure 4.1. The actual architecture of the prototype implemented was a simpler version with minimal semantic matching of user input, identification of simplistic slot-matching, and a response of several pre-programmed outputs. A development user study was conducted using the simplified prototype version of the system [16].

The desire for this system is to incorporate a holistic user experience that allows the user to see a virtual student and their facial expressions and responses as well as use Automatic Speech Recognition and Text-to-Speech rather than typing to simulate a more realistic scenario for the target user interacting with a student. We focus on a multi-modal chat-based interface that allows for user engagement with a diagram referenced and observed by the

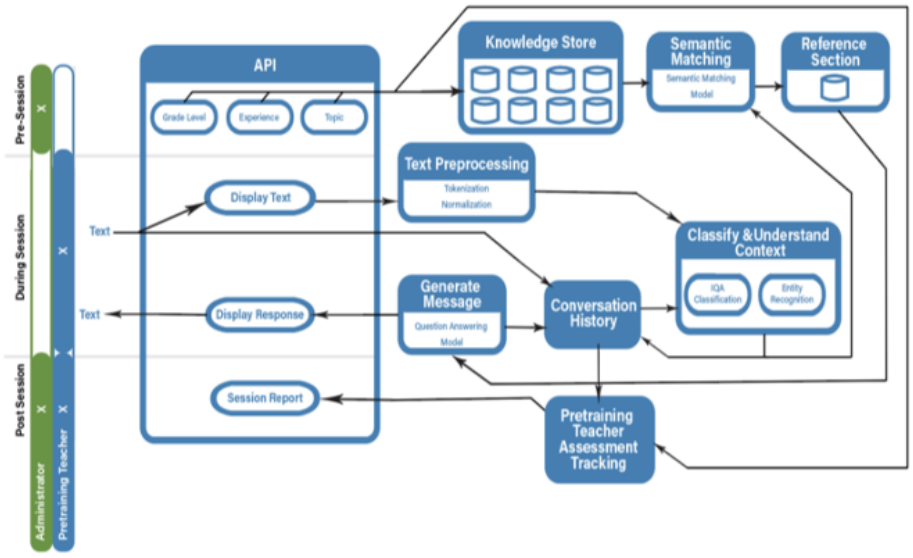



**Fig 4.1.** Proposed Prototype Artificial Intelligence Classroom Teaching System Architecture

virtual student.

We intend to develop a system that allows for multiple experience levels (novice, intermediate, advanced understandings of a topic), multiple scenarios or topics (scale factor, fractions, a science scenario), and multiple grade levels of a virtual student. A user will be able to select the setup before the session. During the session, a user inputs text which is then displayed in the interface and then pre-processed in a Natural Language Processing pipeline.

The text is normalized and tokenized before being tagged for slot references. A slot reference in the case of our prototype would be something like, "What is the length of the right figure?". This would identify the slot length by matching the tokenized and normalized text "length" and "right" to a matching text snippet of code, which essentially targets the user's goal of knowing the value of an entity, the entity being the right figure length. The system would then provide a response with the correct length of the right figure. The conversation history is tracked, and semantic matching is used to identify the slot-filling desire of the user. If the user instead says something such as "Great job!" then that would semantically match to a different prewritten response stored in the database of responses. "Thanks, I

think I'm getting this!". The pre-programmed responses we use semantic matching for are limited in the prototype. We did not utilize the knowledge base semantic matching section and corresponding reference until later in development.

The pretraining teacher assessment tracking comprises an IQA classification count performed on each input text. The input text is classified according to a classifier trained to identify the input category according to IQA, and it counts the number within each category. This is then summed and provided to the user post session. Further development of the feedback system is necessary for the system to provide meaningful output for users.

This concludes the summary of the prototype architecture; in the following sections, additional detail is provided on the development portions. Of note, the code we used in development is Django, and we ran the system on a local computer terminal. For the development user study, the facilitator asked participants to speak about what they would like typed and would type inputs paraphrased for users. The following sections detail the development and consideration of specific components of the prototype.

## 4.3 Knowledge Base Component

At the initial development of the prototype, we did not have labeled knowledge data. Often in conversational agent design, there are FAQ databases or other sources relevant to the development of the purpose conversational agent. As this is the first conversational agent of its kind and with its purpose, there were no readily available datasets. The ACTS architecture incorporates advanced natural language processing capabilities rather than relying solely on simple rule-based structures. Intensive and more comprehensive rule-based architectures are time-consuming to develop, require the expertise of understanding patterns related to the types of interactions a system is designed for, and have high maintenance requirements in extended system use-cases.

### 4.3.1 No Data to Gathering Knowledge Base Data

In the initial ACTS development, we use a retrieval-generative response generation approach for the knowledge base component of the system. This system was developed and not implemented in the first iteration of the architecture deployed as soon after development, the lack of an improved and relevant intent classification component within the dialogue management system was missing. The purpose of this component is to generate responses in the condition when users ask a question of the ACTS related to definitions and textbook knowledge. The pre-processed user input is semantically matched to a knowledge base if this intent is clear. Those two items are provided to a question-answering transformer as input for response generation. The specific question-answering system used in our design is known as a retrieval-generative approach.

Some architectures may incorporate external sources of knowledge which is known as open domain question and answering, [34] as depicted in Figure 2.6. These open domain approaches may utilize sources such as Wikipedia[10]. In developing an open domain response generation component, requirements include large language models containing vast context patterns and referring to the external knowledge base for answers. For the ACTS, this does not meet the need of the type of system desired: a virtual student with imperfect knowledge and anthropomorphic language qualities. A vital component of the ACTS is incorporating a multi-modal approach using a reference image as part of the discussion. This open-domain approach would likely fail at referencing relevant information related to the image included in the scenario itself, and it would fail to appropriately answer questions such as "Can you explain how changing that value to two changes the image?". The knowledge base developed incorporates scenario-relevant video transcripts captured from YouTube and texts from scenario-relevant homework-help websites.

Data is captured in a timely fashion using minimal resources. In Figure 4.2, the implemented data capturing process is illustrated. For the video transcript information capture, relevant YouTube IDs are compiled from a cursory search of the topic on Youtube. Open-

source code converts the IDs to transcripts, and each transcript is then split into single sentences. A domain expert then reviewed the resulting information to verify the appropriateness of the content and whether the information provided aligned with the desired knowledge base content for the ACTS system initial scenario of the eighth-grade scale factor.

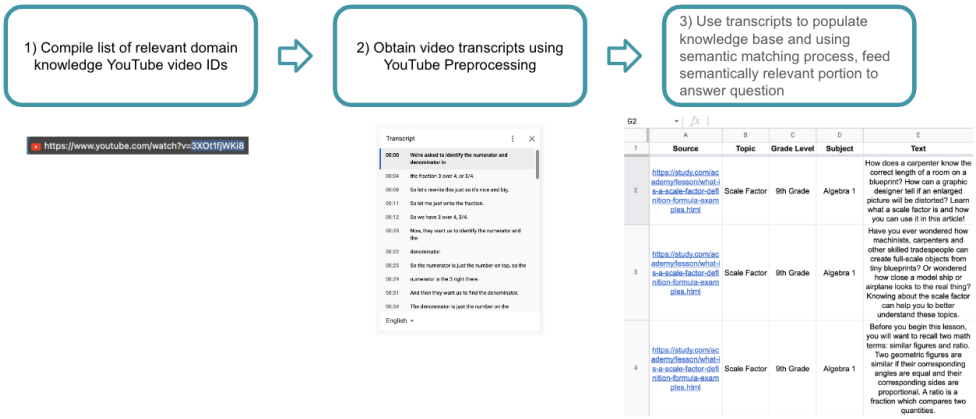



**Fig 4.2.** Starting with No Data: Process to Gather Knowledge Base Data

### 4.3.2 Knowledge Base Implementation: Semantic Matching

The knowledge base of dialogue systems can grow highly complex, depending on the scenario for which the dialogue system is being built[57]. For this initial demonstration scenario, the conversational agent presents as a student with some level of understanding of the topic of scale factor. The knowledge base relies on unstructured knowledge about scale factors collected from the internet and formatted in plain text. ACTS is intended to represent a student's understanding of a topic, which is imperfect; contradictory sources of information are not a primary concern. Contradictory information stored in the knowledge base may be advantageous to anthropomorphically accurate features such as a student's fallibility in understanding. The contradictions can be leveraged to support more robust and realistic answering.

Additionally, a student's expected understanding of a given topic is likely to be documented in instructional materials readily available on the web; therefore, collecting this data

is a simple way to develop the initial iteration of a knowledge base. In collecting this information, domain experts labeled information according to the grade level of understanding and math sub-discipline. Future efforts may incorporate these features in developing more complex interactions with the knowledge base. This may improve the user experience by generating a more realistic student profile with a grade-reflective knowledge base.

The data that was collected was not cleaned or annotated. The labels given to sections of the text are not used for the initial prototype development; therefore, the implementation discussed is unlabeled. Basic pre-processing was implemented to remove figure references and hyperlinks. The text was then separated into sections with no more than 512 words in order to allow the text segment to be used in the chosen text-generation question and answering transformer model.

Relying on unstructured knowledge bases is critical to rapidly developing and deploying new conversational agent scenarios. Unstructured texts on varying mathematical topics are available from websites and video transcripts. Our framework allows a simple way to incorporate newly generated external knowledge bases to scale to additional scenarios. Due to advances in question and answering transformer models and the ease of use of available libraries such as Huggingface Transformers[56], we can use unstructured knowledge and incorporate a retrieval-generative design for response generation. This capability is one step further removed from rules-based and hard-coded responses and aligns with the trend of state-of-the-art capabilities and explorations.

The initial step for the retrieval-generative response component is identifying which segment of information is sent to the question and answering transformer model. To that end, the pre-processed input text is combined with the Universal Sentence Encoder [8] to find the knowledge base's most relevant or semantically similar section. The term or method of semantic similarity is a method that identifies the degree to which two texts have the same meaning. As earlier mentioned, the plain text in the knowledge base is divided into segments of 512 characters or less as that is a requirement of the transformer input text length for

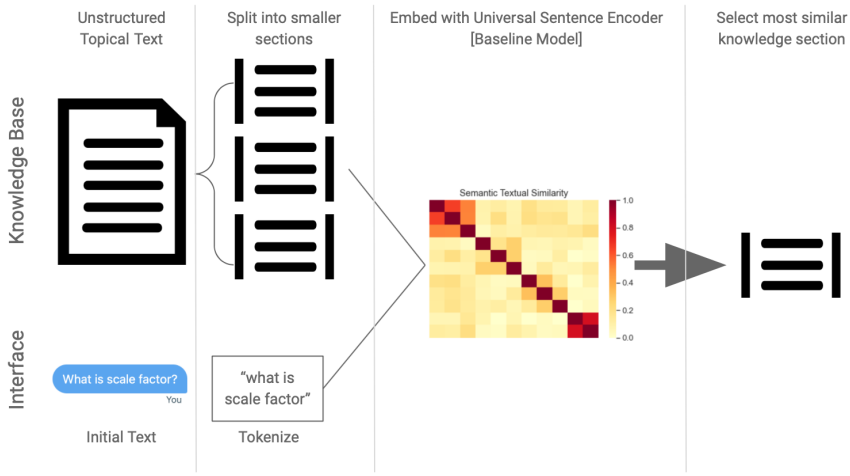



**Fig 4.3.** Retrieval-Generative Response Generation Component for Knowledge Base Queries Utilizing Semantic Similarity

the model implemented. This is also the size limitation for the semantic similarity model implemented: the Universal Sentence Encoder. I depict this process in Figure 4.3.

The Universal Sentence Encoder is optimized to read in short phrases or paragraphs and outputs a 512-dimensional vector. Semantic similarity computation is achieved by computing the inner product between the input and knowledge base text. We computed the semantic similarity between generated embeddings using normalized cosine similarity. Semantic similarity computation by sentence is more accurate than the aggregate of word-level similarities and is therefore preferred in this application. Models trained to understand words in a more holistic context are often better suited for identifying semantic similarities between phrases and sentences. In application, we can expect input such as "What is scale factor?" By finding the most semantically similar section in the knowledge base, we can use this section to input the response generation.

A pre-defined threshold is set, and the semantic similarity of each knowledge base section and the user input text is captured. The top scoring semantic matches are obtained; if the threshold is not met, then the system logic would be sent to respond with a random pre-determined response indicating that the conversational agent does not know the answer. For the prototype development, the threshold was set to 0.80 after empirical evaluation of the semantic coherence. I recommend for future development and subsequent system iterations

to test and evaluate the threshold value.

The pre-determined responses indicating that the system does not know the response are a simplistic set of six prewritten variations of semantically similar, developer-crafted text of "I don't know". These responses exemplify the hybrid nature of the overall conversational agent architecture. The ability to move to fully generative responses is not feasible given state-of-the-art Natural Language Processing methods' resource limitations and technology advancement constraints. This is a hand-crafted rules-based response that does not incorporate generative technologies. It is essential to this nascent development of a conversational agent system in a low data and resource environment.

If, on the other hand, the semantic similarity is greater than or equal to the threshold for a given knowledge base section, the system then selects this knowledge section. The initial input text and the selected knowledge base section are used as inputs to a question-answering module. The question-answering module is a pre-trained BERT model that is fine-tuned on the Stanford Question Answering Dataset (SQuaD). We use the fine-tuned pre-trained BERT question and answering model to generate a response to send back in the user interface for the subsequent conversation turn. Dialogue states of the conversation are recorded for reference within the conversation. The prototype uses a simplistic state tracking of slots filled for dimension values of the image and the scale factor value. An example of the prototype slots implemented within the dialogue management of the system is shown in Figure 4.4. The logic to track these slots was simplistic text matching code to identify numbers and slot references, and it required users to directly list elements of the slot names in specific ways in order to access or update information. For example, this required users to explicitly state "left_figure" as opposed to free-form natural referencing such as "the figure on the left". This simplistic matching is expanded upon in subsequent iterations of the system.

{"experiment_id": "12", "left_figure_length": 0,
"left_figure_width": 0, "left_figure_height": 0,
"left_figure_volume": 0, "right_figure_length": 0,
"right_figure_width": 0, "right_figure_height": 0,
"right_figure_volume": 0, "scale": 0}

**Fig 4.4.** Prototype Artificial Intelligence Classroom Teaching System (ACTS) Entity/Slot Tracking

## 4.4 Skills Feedback Mechanism: Instructional Quality Assessment

The Instructional Quality Assessment(IQA) has been developed by the Learning Research and Developmental Center at the University of Pittsburgh since 2002 [40]. The IQA provides a comprehensive assessment of mathematical instruction and can be used in assessing the academic rigor of discussion surrounding a task [46]. The IQA has since been further validated in subsequent research by its original developers [6]. In recent literature, researchers suggest the IQAcan be implemented as a support tool for pre-service or in-service teachers to receive feedback or meaningful assessment, which may help improve their instruction [6].

Teachers' questions are essential for their students' growth in meaningful mathematical discourse. The academic rigor component [27, 4] of the IQA builds upon earlier classifications of teacher questions (e.g. In the literature, [1] there is a distinction between a "probing and exploring" question which is a user input that is intended to invite students to clarify their ideas and the relationships between them, and a "procedural and factual" question which is meant to elicit a fact or yes or no response). The IQA was developed for use in contexts with cognitively demanding mathematical tasks. It is well suited for the professional development of pre-service and in-service teachers, especially given those parts that focus on teacher questioning [5].

Teachers' questions are essential for their students' growth in meaningful mathematical discourse. The academic rigor component of the IQA builds upon earlier classifications of teacher questions (e.g. The classifier can be used to distinguish between a "probing and

exploring" question which is a user input that is intended to invite students to clarify their ideas and the relationships between them, and a "procedural and factual" question which is meant to elicit a fact or yes or no response). The IQA was developed for use in contexts with cognitively demanding mathematical tasks. It is well suited for the professional development of pre-service and in-service teachers, especially given those parts that focus on teacher questioning.

Research has shown that the Instructional Quality Assessment provides a robust framework for evaluating teachers' instructional practice in mathematics classrooms. Therefore, we have selected this framework as the basis for our feedback component. The framework we craft is a modified interpretation of the IQA constructed to assess one-on-one discussion through a web-based platform. The modified IQA classifier is intended to provide feedback on the quality of questioning strategies for pre-service and in-service teachers. Although the modified IQA is the framework for the feedback mechanism implemented in the prototype and subsequent ACTS development in the context of this dissertation, alternative feedback or assessment component could be constructed and integrated into the system with relative ease.

The Instructional Quality Assessment (IQA)[4] is a well-established framework for evaluating mathematics instruction. We developed a system for pre-service teachers, individuals in a teacher preparation program, to evaluate teaching instruction quality based on a modified interpretation of IQA metrics. Our demonstration and approach take advantage of some of the most recent advances in Natural Language Processing (NLP) and deep learning for each dialogue system component. We built an open-source conversational agent system to engage pre-service teachers in a specific mathematical scenario focused on scale factor, intending to provide feedback on pre-service teachers' questioning strategies. We believe our system is practical for teacher education programs and can enable other researchers to modify it, building new educational scenarios with minimal effort.

**Table 4.1.** Instructional Quality Assessment (IQA) Modified Categories

| Question Label | Examples |
|---|---|
| Probing or exploring | • How did you get that answer?<br>• What does n represent in the diagram?<br>• Why is it staying the same? |
| Factual or recall | • What is 3x5?<br>• Does this picture show 1/2 or 1/4?<br>• What do you subtract first? |
| Expository or cueing | • Rhetorical questions ("The answer is _, right?")<br>• Clarifying statements "Between the 2?"<br>• Look at this diagram |
| Other | • Sit down<br>• Close your books |

### 4.4.1 Classification Overview

We fine-tune BERT [17] and DistilBERT [50] for classification of the IQA based on the open source Huggingface Transformers implementation[56]. We used 80% of the data to train the classifiers and sectioned the remaining data as a 10% validation set and a 10% test set. We achieve an accuracy of 75.8 for the fine-tuned BERT model and 74.3 for the fine-tuned DistilBERT model. The difference in accuracy of 1.5 is minimal, and the speed advantage of deploying and incorporating the DistilBERT model resulted in the use of DistilBERT in the prototype development.

The feedback mechanism incorporated in the conversational agent design of ACTS is critical as the objective of this meta-purpose agent is to provide feedback on questioning strategies of in-service and pre-service teachers in their one-on-one student interaction skills development. The prototype implementation of ACTS classifies each user utterance with the modified IQA categories. The categories of the modified IQA measure were developed through a joint effort with education domain experts, and the categories were iterated over several months during the classification and labeled data development process.

Table 4.1 outlines the categories defined through the collaboration of domain experts.

### 4.4.2 Classification Process

Annotators also had the opportunity to flag any data as a "data issue," representing a transcript pre-processing error or another issue (i.e., blank or incoherent) indicating that the data could not be labeled.

The data used with the modified IQA evaluation rubric was developed from transcriptions of audio recordings of teachers in whole-class and teacher-student conversations in elementary mathematics classrooms using different mathematics curricula across the United States. The de-identified dataset was shared from an NSF-sponsored project that had previously collected the recordings to answer separate research questions. In the recordings, students engaged with a project purposed to help them understand different geometry concepts like scale factor, dimensions, surface area, and volume of rectangular prisms. The students recorded their observations from a given visualization and explained the impact of the scale factor. The data collected for the development of this scenario contained 2826 questions. The unique question and the context, or speaking turn in which the question was uttered, were both provided as references for the annotators to use during labeling.

We had 5799 total labeled data instances. There were five total annotators: three expert teachers(defined as teachers with at least several years of experience) and two pre-service teachers. The total number of annotators fluctuated during different stages of the annotation process resulting in varying amounts of labels generated by each annotator. The time to label each data point averaged between 5.2 to 6.7 seconds per annotator. The total number of unique labeled sentences was 2826. The total distribution of labels between the four assessment categories ranged from 856 to 2133. We used weak supervision-based approaches to combine the labeled data from multiple annotators over majority vote approaches.

### 4.4.3 Efficiency Labeling

Given the limited time domain experts may have for annotating data, we explored several methods to improve label efficiency. As highlighted in the literature [48], weak supervision

techniques provide the two-fold benefit of requiring less human labeling than would otherwise be required for training. An additional benefit of weak supervision is that noisy data and each annotator's accuracy can be considered for classification. The literature[48] shows that weak supervision systems are better than generic majority vote approaches. Noisy label data for model classification has also been studied in deep-learning-based approaches [20] and proven effective.

#### 4.4.3.1 Labeling Method

Two labeling platforms were used extensively for this project: Labelbox and Label Studio. While both platforms were straightforward, Label Studio allowed custom user interfaces with several improved features, such as keyboard shortcuts that allowed annotators to onboard and complete labeling tasks more efficiently.

Each question was labeled with a context reference that allowed annotators to see the entire speaking turn of the teacher. We decided to include context after observing how previous iterations of labeling questions resulted in an inter-annotator agreement of below 0.50, which subsequently increased to 0.66 after including context.

Our data collection approach relied on weak supervision and learning with noisy label strategies. Noisy labels acquired in this paradigm, either through human labels or machine learning models, are cost-effective to acquire. When domain expert annotators are available (in our case, expert teachers), noisy disagreements between annotators can be leveraged to build high-accuracy models [48, 20]. Weak supervision approaches are scalable, enabling easy adaptation to multiple mathematical scenarios, one of this project's critical contributions and focuses.

### 4.4.4 Weak Supervision

We experimented with multiple text classification approaches, including Convolutional Neural Network (CNN)—based text classification [28], Long Short-Term Memory (LSTM)—based text classification [35], and newer approaches that rely on Transformer Architectures

**Table 4.2.** Weak Supervision Results

| Technique | $n$ | Acc. | Agreement | Time | | Transfer learning |
|---|---|---|---|---|---|---|
| | | | | M(SD) | p-value | F1 score |
| Classical Labeling | 1730 | 0.82 | - | 15.2 (42.1) seconds | <0.001 | 0.712 |
| Weak Supervision Model-Assisted Labeling | 3983 | 0.84 | 0.7 | 10.4(32.7) seconds | | |

[17, 36] and perform well with small amounts of labeled data. Transfer learning models tend to perform well with less labeled data than other models because of the pretraining with unsupervised text that encodes knowledge and semantic meaning of words and sentences. This demonstration incorporates a fine-tuned Bidirectional Encoder Representations from Transformers (BERT) model for our modified IQA classification task.

### 4.4.5 Codifying Data Labeling Process

An overview of the process is provided in Figure 4.5. A brief outline of this process is summarized as follows: I propose first gathering data. In this process, our team initially used transcripts; however, in a later iteration, I utilize generated data. With weak supervision, our team used a system called Snorkel to incorporate noisy data and develop a classifier. Weak supervision is best used when there is a clear signal, but there is no clear agreement within a label; an example is if multiple labelers classify the same data with differing results. Weak supervision can be used to find the underlying signal and develop a classifier that still works well.

With the noisy labels of data, we use Snorkel to perform weak supervision; we then send the resulting information to a fine-tuning pipeline to leverage a pre-trained transformer and fine-tune it for our classification purposes. Through the Huggingface pipeline, we can fine-tune efficiently and with little coding expertise required. We then can utilize the classification model on additional labels: when the confidence of a classification is uncertain, we use active learning to allow a human to verify classification categories for those data labels. This process

can be repeated multiple times as additional data is gathered. The entire classifier developed can then be used for alternative scenarios, one example of implementing transfer learning. [15]

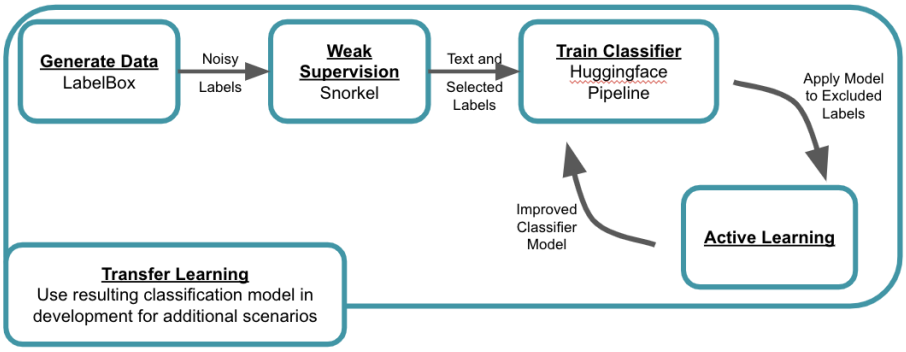



**Fig 4.5.** Process to Develop Labeled Data Efficiently

### 4.4.6 Dialogue State Tracking

Dialogue State Tracking is a core component of the dialogue system. The goal of the dialogue state tracking system is to interpret the user's purpose within each turn of the conversation. There are multiple formulations of dialogue state tracking systems, like hand-crafted rules [53] or a web-style ranking [55]. In this prototype version of our system, we use a question and answering paradigm for response generation [19]. Unlike [19], we do not train our retrieval-generative based model. Instead, we use the question and answering paradigm to support the logic behind the dialogue states.

Further logic integrating the modified IQA classifier and the identified entities must be designed and developed. This is not a task-specific dialogue system; simply identifying the entities' state at the end of a session does not indicate a successful system. This is a distinction between a customer-service agent and a meta-purpose agent. In the subsequent iteration of the system, the dialogue management system and state tracking are expanded and detailed.

An example of the interface, as well as a demonstration of each user utterance classifica-

tion by the IQA, is depicted in Figure 4.6.

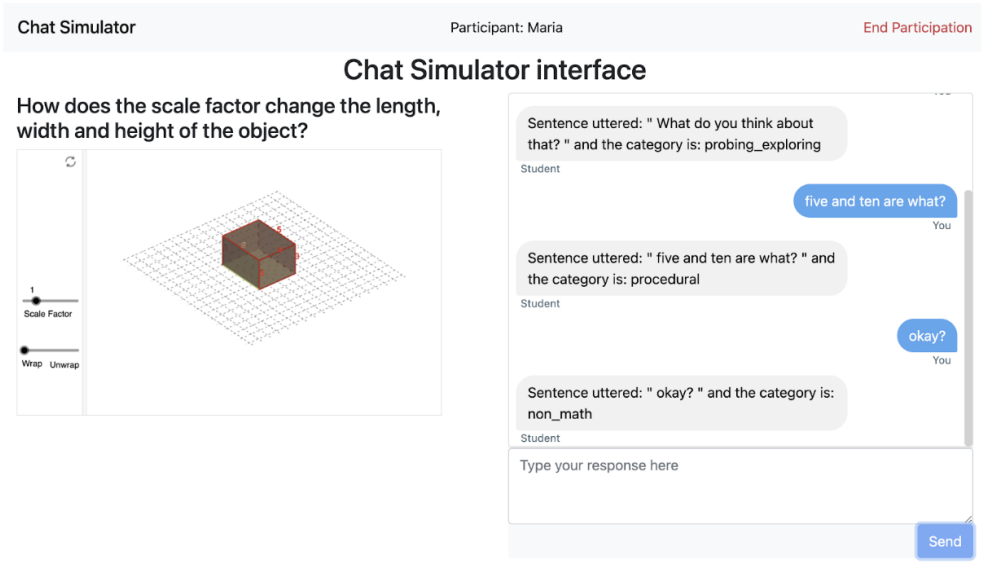



**Fig 4.6.** Prototype Example Interface: Classification of User inputs by IQA Category

### 4.4.7 Response Generation

The response generation component extracts relevant sections of the knowledge base as part of a question-answering task. A question-answering task is a supervised learning problem where given a segment of text of $i$ tokens and a question of $j$ tokens; it returns an answer segment of $k$ tokens. The answer in question-answering tasks can be cloze-style, as in CNN/Daily Mail [21], span prediction (like SQuaD [47]), or be similar to Narrative QA. We retrieved our knowledge from semantic matching of web-text categories and thus our response generation pipeline matched closely to span prediction tasks. We implemented the response generation pipeline using the transformers library [56], where a BERT model [17] was fine-tuned on the SQuaD dataset. We did not fine-tune our question-answering system for the response generation module. Instead, we relied on semantically-matched, unstructured data sections to be used as inputs in generating answers to questions.

### 4.4.8 Session Feedback

All text input by the pre-service teacher and the associated adapted IQA category classification is retained. The compilation of classifications of the pre-service teachers' input texts

is captured and can be compiled in a post-session assessment report. Implementing the IQA classifier is the first step toward more meaningful feedback provided to users after engaging with the system.

## 4.5 Prototype Additional Component: Ozchat

Our intent while developing the prototype is to gather usability information to inform improvements in future design iterations of the system. One well-documented way of doing this is by incorporating an "Oz" component. An Oz component is when a system has the capability to allow a facilitator or administrator to interject in place of the conversational agent. This is key during development as building out the dialogue management system and a robust dialogue system without increments is infeasible. The Oz component allows for a session to continue without breaking down completely, all while continuing to gather information when the conversational agent may have otherwise not responded appropriately or at all.

In our system, we incorporate several versions of Oz - one option allows for simple observation of a conversation, a second option which allows for the facilitator to step in if the conversational agent is sufficiently confused by a user's input, and a third option in which the facilitator does all the interacting in place of the conversational agent.

## 4.6 Conclusion

My goal in this chapter is to demonstrate and outline the process of developing the prototype for the ACTS system. I, along with the ACTS team, implemented a conversational agent prototype with very little training data that incorporates a well-studied feedback metric, the IQA. We built a functional prototype by leveraging state-of-the-art modules for natural language processing and deep learning. By integrating pre-trained models such as SQuAD, BERT, and the Universal Sentence Encoder and using weak supervision approaches in data treatment, we have leveraged minimal amounts of domain-expert-labeled data and knowledge base data to create a usable interface.

The following research questions were addressed in this chapter:

**Research Question 4.1**: *Can we design a system that fills pedagogical needs for individual skills practice, modernizes conversational agent approaches within teachable agents, and adopts a meta-purpose framework?* We develop a novel pedagogical teachable agent that incorporates modern NLP technologies and user experiences, addresses the digital system gap in pedagogical individual skills practice digital systems, and contributes to the diversification of conversational agent design in the framework of a meta-purpose agent.

**Research Question 4.2**: *How can we overcome low-to-no-data and develop critical components of conversational agents such as a knowledge base?* We propose a pipeline for establishing a knowledge base in a no-data scenario and discuss implementation utilizing accessible NLP technologies and frameworks such as Information Retrieval, Retrieval-Generative Response Generation, and Semantic Matching.

**Research Question 4.3**: *Can we develop a framework for conversational agent classification-element-development in low to no data scenarios?* We propose a novel pipeline to maximize resources available in low-to-no data scenarios when creating classifiers, thereby mitigating the challenges of developing or acquiring large labeled data sets. We validate our proposed pipeline and demonstrate the significant efficiency of this methodology.

# Chapter 5

# No Data Dialogue Management Development

## 5.1 Background

In this chapter, I discuss in detail the iteration of a further developed dialogue management component of the Artificial Intelligence Classroom Teaching System (ACTS). This iteration followed the prototype version and incorporated more advanced logic and capabilities. I address the following research questions:

**Research Question 5.1**: With insights from preliminary testing, what changes can I implement to improve the design of a pedagogical teachable agent? Can these improvements be demonstrated to allow for transparency in the development of a conversational agent process? *This addresses a gap in the literature for Conversational Agent Gap 3 in Section 2.1.6.*

**Research Question 5.2**: In a no data, minimal time scenario, what is an effective way to deploy a new intent classification component within a conversational agent design? *This addresses gaps in the literature for Conversational Agent Gap 1 in Section 2.1.6, Conversational Agent Gap 2 in Section 2.1.6, and Niche Gap 1 in Section 2.3.*

**Research Question 5.3**: When building a dialogue management system, how

can I utilize a generalizable structure to minimize the requirements in developing additional scenarios? *This addresses gaps in the literature for Niche Gap 1 in Section 2.3.*

## 5.2 Architecture Approach

As discussed in Chapter 2, there is a lack of generalizable conversational agent design within the education domain. I focus a primary design element to be modular rather than end-to-end, and I include intentional effort to avoid hard-coding scenario features within the system. The IQA feedback classification is a generalized model that applies to multiple teaching scenarios to provide feedback for questioning skills in mathematical and science-based scenarios. I can reuse that component without having to change other elements within the system.

### 5.2.1 Generalizable Design and Entity Development

One element of a generalizable design is a generalizable framework. I incorporated the use of dynamic variables within the code and conceptualization of design scenarios to emphasize this quality.

In order to achieve this, I deconstruct educational scenarios by the elements, thereby providing a common language to communicate between the conversational agent and the user. I establish four categories of variables and refer to them as Primary, Secondary, Independent, and Formulas. The implementation of this approach in ACTS is depicted in Figure 5.1. The primary dynamic variables are the "green" and "blue" figures, the secondaries are the dimensions and key values such as the volume or surface area, the independent dynamic variables are the scale factor and the units, and the formula captures mathematical relationships between the dynamic variables such as a volume formula. These elements are all coded in a way where the developer can define the dynamic variables and automatic entity recognition and tracking is implemented within the code.

This approach incorporating dynamic variables is generalizable beyond the scale factor

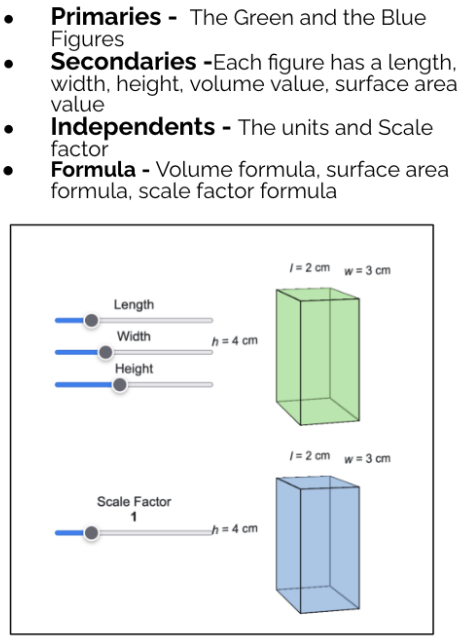



**Fig 5.1.** Generalizable Entity Development

scenario that is currently developed and I review each category utilizing an alternative scenario where this could be applied as well shown in Figure 5.2. Primary variables describe the "objects" or "entities" in a given scenario. In Figure 5.2, you can see two fractions – thus, there are two primaries in this scenario, and they might be referenced with phrases like "the left fraction," "Fraction A," or "the green fraction." Next, secondaries. Secondary variables refer to the fields that each primary variable has - to refer to Figure 5.2 again; any fraction will have a numerator and a denominator. Since each primary has its own set of secondaries, in this example, there are two secondaries (numerator and denominator) but four unique secondary values. Independent variables are values of which only one will ever be for the whole scenario. The sign (greater than, equal to, less than) represents independent variables depicted in Figure 5.2.

Moreover, formulas are variables that represent the system's method of calculating any other variable on the fly. While my fractions example does not have any of the classic formulas like "volume" or "surface area," I would still implement it with formula variables to calculate the fraction's value and determine if it is equivalent to another fraction. This common

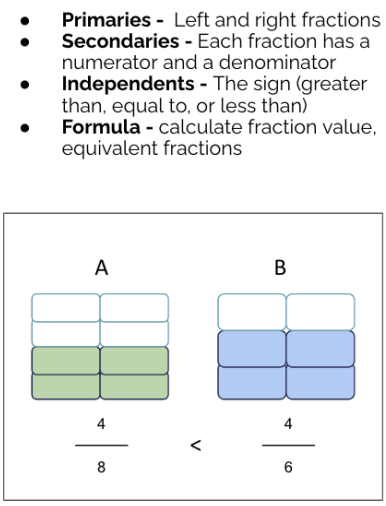



**Fig 5.2.** Generalizable Entity Development Example of Transferability

framework allows for dynamic variable coding and using set values in entity recognition and other dialogue system components.

### 5.2.2 Intent Categorization for Virtual Students

Moving on from dynamic variables, I propose new intent categories not previously found in the literature. In the literature, the majority of intent discussions are oriented around customer service solutions for customer-facing conversational agent development. I develop virtual student conversation related intention categories to better identify user intent from a teachable agent design perspective. There are several categories and subcategories. The main intent classification categories are:

- Connect : Utterances intended to build connection and rapport or polite greetings with the student
- Pump : Asking the student information (Value, yes or no, clarification)
- Inform : Providing information to the student
- Feedback : Providing positive, neutral, or negative acknowledgment of student direction

- none : not pertaining to the problem material

### 5.2.3 High-level Architecture

These intent categories fall within a greater architecture construction alongside the previously discussed entities. The overarching architecture is depicted in Figure 5.3.

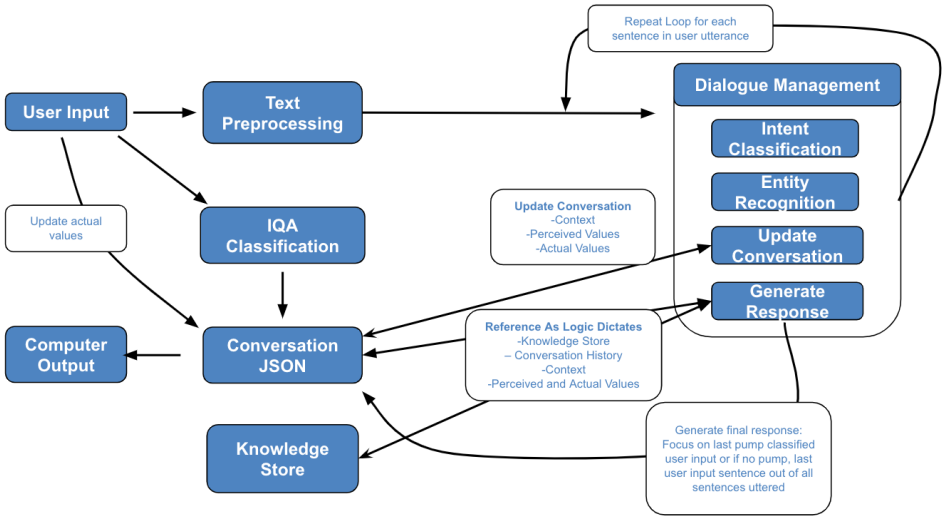



**Fig 5.3.** Overarching Framework

A vital component of this process is the natural language process underpinnings—an example of the natural language pipeline implemented through the Spacy library. I use a standard pipeline with few alterations to allow for input text tokenization, lemmatization, and intent matching via a Spacy parser.

Tokenization is the process of taking a text that requires analysis and breaking it down into "blocks" called tokens that the conversational agent can begin to make sense of.

Lemmatization is the next step of the natural language pipeline, in which the conversational agent begins to take differently formatted yet similar words and group them together. Roughly, the system begins making categories or buckets for tokens/words expressing the same ideas.

Intent matching continues the journey of natural language understanding by taking the lemmatized tokens and attempting to conclude what the input text is "intending" - what are the speaker's goals or desires, as expressed in the text being read?

## 5.3 Detailed Intent Architecture Logic Discussion

To understand precisely how the system can take in user input and generate a realistic response, I walk through all of the different intent categories and see how the system is designed to respond in each case. The following sections will each offer a diagram, and a text section will elucidate how the system keeps track of the state of the conversation and responds as realistically as possible.

### 5.3.1 Primary Intent Connect

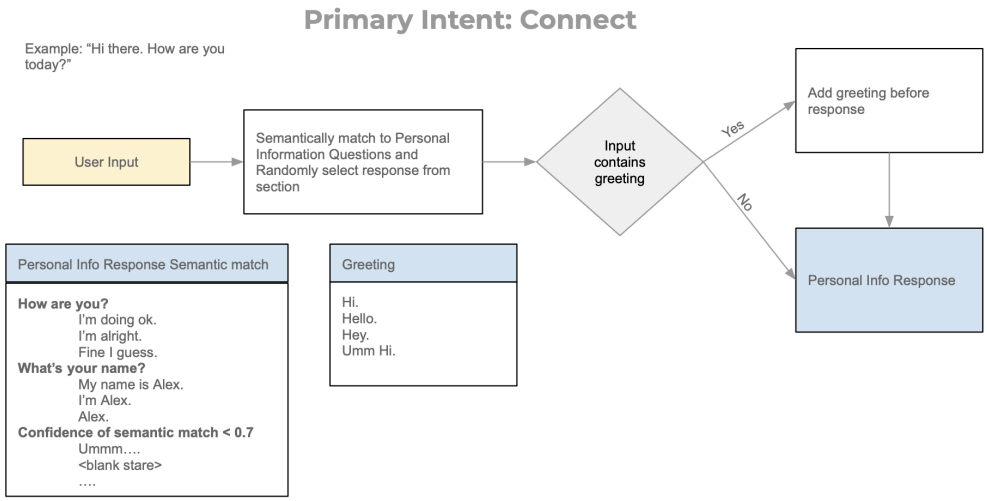



**Fig 5.4.** Logic Diagram for User Input Classified as "Connect"

Figure 5.4 shows the logic flow as the system responds to a connect statement. It takes the user input and uses the Universal Sentence Encoder to search for a semantic match in a list of personal questions. If the confidence threshold is met, the system prepares a randomly selected response to that question. If the original input contained a greeting like "Hi" or "Hello," a return greeting is added to the beginning of the response.

### 5.3.2 Primary Intent Pump

Figure 5.5 shows how the system would attempt to respond to a "testing" question. The logic flow of this attempt goes something like this: it starts with a user's input (i.e., "How would you calculate scale factor?"). Using the Universal Sentence Encoder, the system will take that user input and search its prior session knowledge for a semantic match above a given

**Table 5.1.** Intent Category Descriptions for Initial Iteration of System

| **Main Intent** | **Sub Intent** | **Description** | **Example** |
|---|---|---|---|
| Connect | | Beginning an interaction or building a connection with a student external to direct education goal | *Hi, how are you today?* |
| Pump | Value | Asking the student to provide some value response | *Can you tell me what the length is for the green box?* |
| | Clarification | Asking for the student to provide further information to a previous response. | *Which object is bigger?* |
| | Testing | Testing student understanding of a topic. | *Can you tell me what a reduction is?* |
| Inform | | Provide information to the student. | *The scale factor is now 2.* |
| Feedback | Positive Feedback | Encouragement that the student is correct with no suggestions | *Yes, that's right. Good Job.* |
| | Neutral Feedback | Acknowledging a student utterance with no indication of direction. | *Ok, I hear you.* |
| | Negative Feedback | Indicating that the student's understanding is not correct or the student needs to change direction. | *I see where you're going with that but that's not exactly the full picture.* |
| None | | The user input is either not relevant, or it is not clear | *Do you think the playoffs game will go into overtime?* |

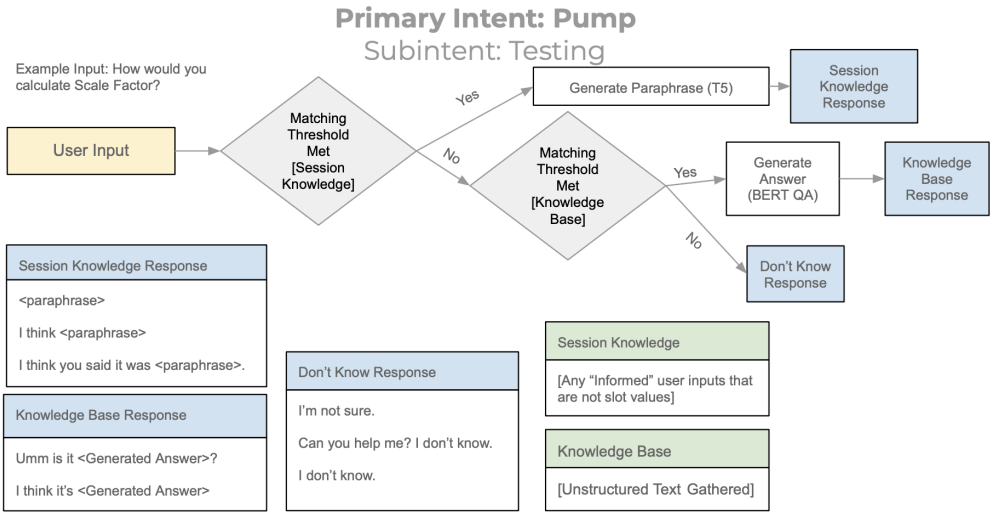



**Fig 5.5.** Logic Diagram for User Input Classified as "Pump" with a Subintent Classification of "Testing"

confidence threshold. If the threshold for a successful match is met, then it will generate a paraphrase using T5 and return a "Session Knowledge Response" - for instance, "Umm, I think that you said earlier that scale factor was the number by which the first figure gets multiplied." However, if that threshold for a match is not met, the system instead turns to its knowledge base. If a threshold for semantic matching is found there, then the system uses BERT QA to generate its response, like "Well, I think scale factor is the ratio between two figures." If the threshold for a semantic match is not met in the session knowledge or the knowledge base, then the system turns to its final default, a "Don't Know" response. This is as simple as, "I don't get it, can you help me?"

Figure 5.6 offers another example of the system in action. This example takes a user's input: "What do you mean it gets bigger? What gets bigger?" The system correctly concludes that this text has a "pump" intent, asking a question of the system. It also concludes that the subintent is "clarification" - the user is asking the system to explain or expand on its last statement. To avoid a cycle of continued user clarification questions with system-generated identical responses, the system generates a random number to simulate a "decision" between two paths, allowing for various responses. On the first path, it takes the text that the user is asking for clarification on, and it paraphrases that text using T5 to produce a reworded

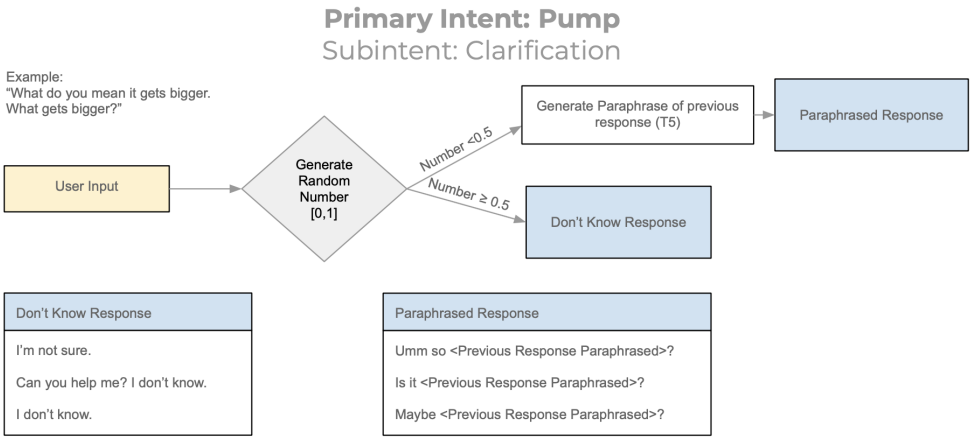



**Fig 5.6.** Logic Diagram for User Input Classified as "Pump" with a Subintent Classification of "Clarification"

restatement. This might be something like, "Umm, the blue figure gets bigger." The system might provide another "Don't Know" response on the second path, like "I'm confused."

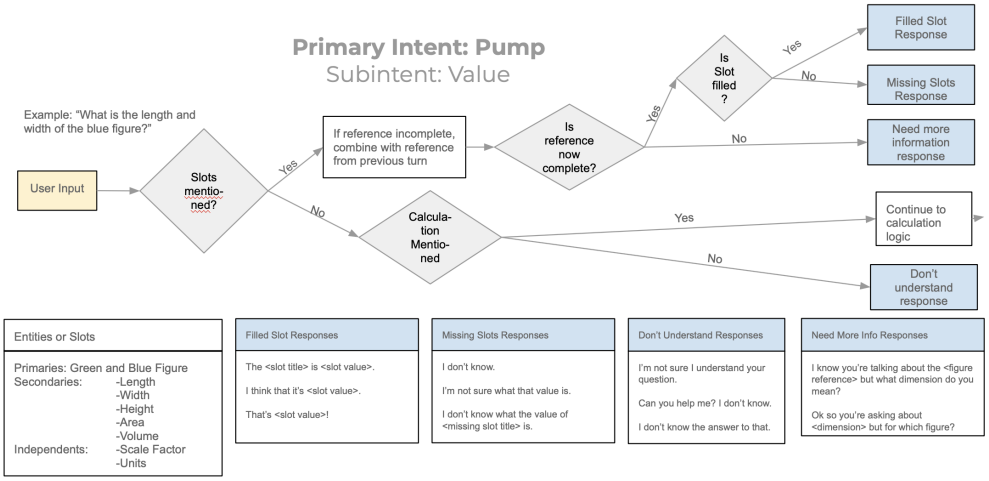



**Fig 5.7.** Logic Diagram for User Input Classified as "Pump" with a Subintent Classification of "Value"

Let us review one more example of the system figuring out how to reply to a given intent. In Figure 5.7, the user input will be a pump-value intent. The system can tell that it is asking a question (pump) and that the question is for the numerical value of a field in the scenario (value). The system will immediately attempt to identify what "slots" the user input is referencing. A "slot" can be considered a fully defined reference to one concept in the problem with a numerical value. Internally, the system determines these slots by using

the dynamic variables mentioned earlier. A slot also called a complete reference, would be any combination of primary and secondary values. In the system, the primary variables are the two objects, and the secondary variables are the fields like length, width, or height so that the slots would be combinations like "the blue figure's length" or "the right figure's volume." The shorter path of logic is the scenario in which the user requests a value but does not even give a partial reference to a slot. This could occur if the user requests that the system calculate a value, so the system will check for that condition and then return a "Don't understand" response if this is not the case. If there is at least a partial slot, the system logic continues forward. To do so, it needs a completed slot. If the user input only contains a partial slot (i.e., "What about the height?"), then the system can look at the previous turn for context. If the reference is complete from the user's input, or if the system can create a complete reference by referring to the last turn, then the system is ready to check if it knows the requested value. If it has a known value for that slot, it will reply with it - "The blue height is 3." Otherwise, it will give a "Don't know" response. If, however, the system cannot complete the reference at all, even after referring to the prior turn, it will have to request clarification - "Okay so you're asking for the height, but for which figure?"

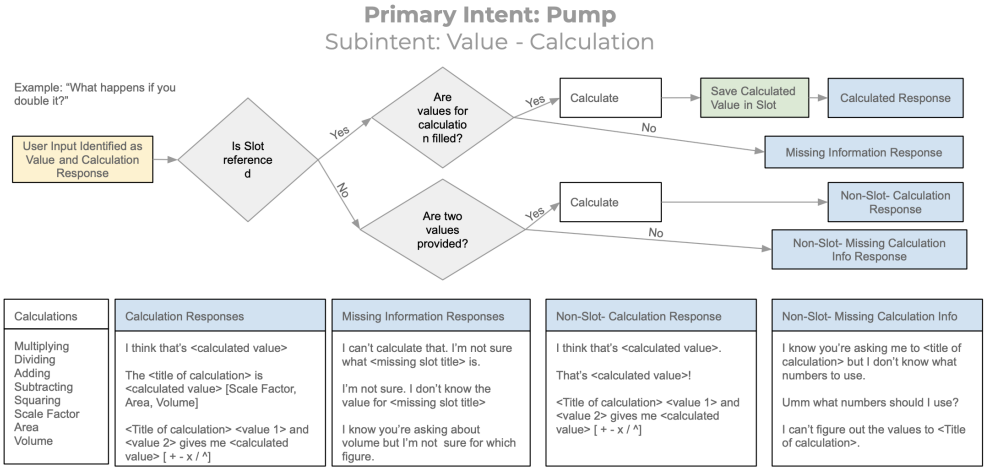



**Fig 5.8.** Logic Diagram for User Input Classified as "Pump" with a Subintent Classification of "Value" and an Identified Calculation

The final variation of pump that requires inspection is that case where a calculation is requested, i.e., "What happens if you double that?" If a slot is not referenced, the system

checks to see if the numbers for calculation are provided. If so, the system can efficiently perform the calculation and reply with the answer. "Okay, I think that 2 + 3 is 5." If no slot is referenced and the numbers for calculation are not provided, the system offers a reasonable reply indicating they cannot perform the requested operation. "I know you're asking me to multiply, but I don't understand what numbers I need to use." If, on the other hand, the requested calculation references a slot (i.e., "What is 2 times the blue length?"), the system checks to see if it has a value for that slot, and if it does, it saves the newly calculated value into the slot and replies as before, "I think that's 18." If the system does not know the value for a slot, it replies with a variation of "I can't calculate that, I don't know what the blue height is."

### 5.3.3 Primary Intent Feedback

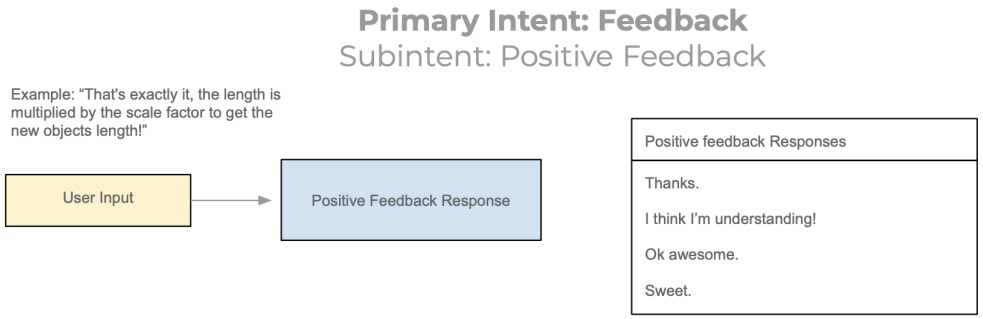



**Fig 5.9.** Logic Diagram for User Input Classified as "Feedback" with a Subintent Classification of "Positive Feedback"

In Figure 5.9, we see the fairly simple logic flow when the user inputs positive or encouraging feedback - the virtual student selects randomly from a set of reasonable responses, like "Thanks" or "Sweet."

Neutral feedback (shown in Figure 5.10) is the feedback that acknowledges a system response with no encouragement or discouragement. It is handled similarly to positive feedback, simply with a different set of randomized response options.

Negative feedback (shown in Figure 5.11 follows the same pattern as positive and neutral, once again just with a new set of possible responses.

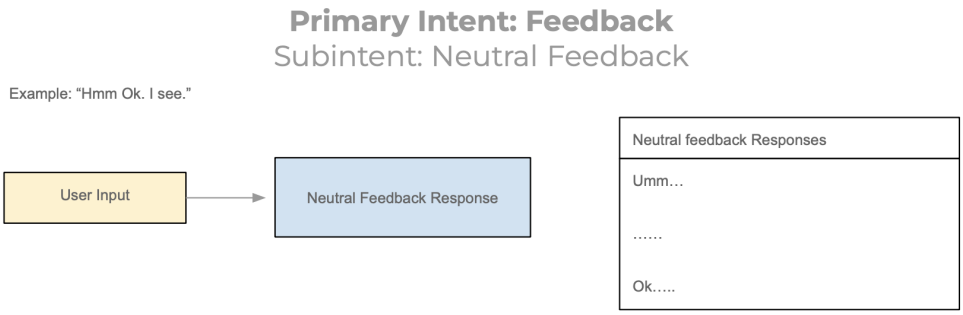



**Fig 5.10.** Logic Diagram for User Input Classified as "Feedback" with a Subintent Classification of "Neutral Feedback"

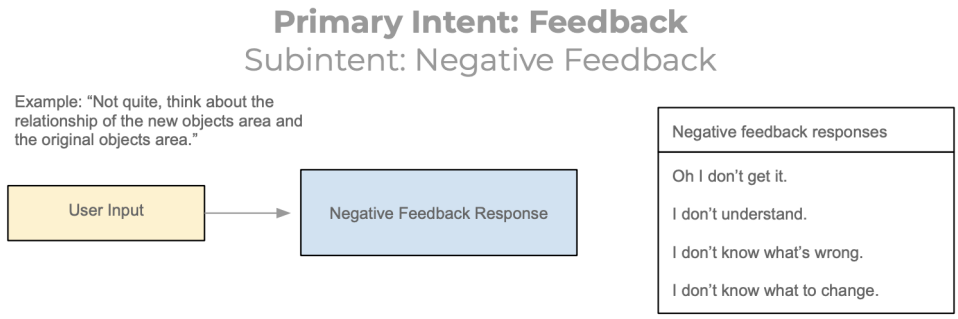



**Fig 5.11.** Logic Diagram for User Input Classified as "Feedback" with a Subintent Classification of "Negative Feedback"

### 5.3.4 Primary Intent Inform

When a user sends a text with the "inform" intent, there are two main options, as you can see in Figure 5.12. On the first path, the informing text references a slot and provides a specific value (i.e., "The green volume is 24"). In this case, the system saves that value into the appropriate slot for future use and acknowledges the change ("Got it, the green volume is 24"). If the new information is conceptual or otherwise not referring to a slot (i.e., "you need to multiply by the scale factor instead of adding it"), that information is added to the session knowledge list, and the input is paraphrased and returned as acknowledgment ("Ok I see, I need to multiply by the scale factor").

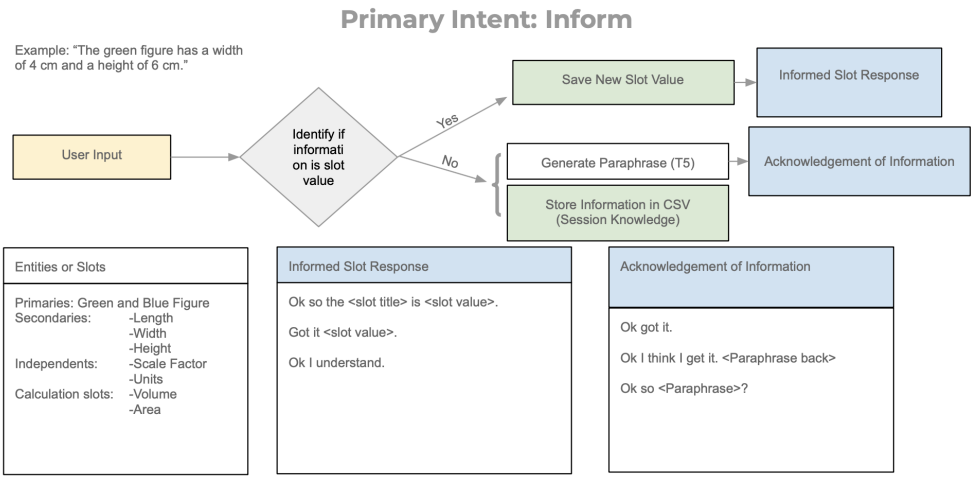



**Fig 5.12.** Logic Diagram for User Input Classified as "Inform"

### 5.3.5 Primary Intent None

As the check for input unrelated to the problem, the "None" intent will earn a response from the system expressing confusion. Figure 5.13 lists some possible responses.

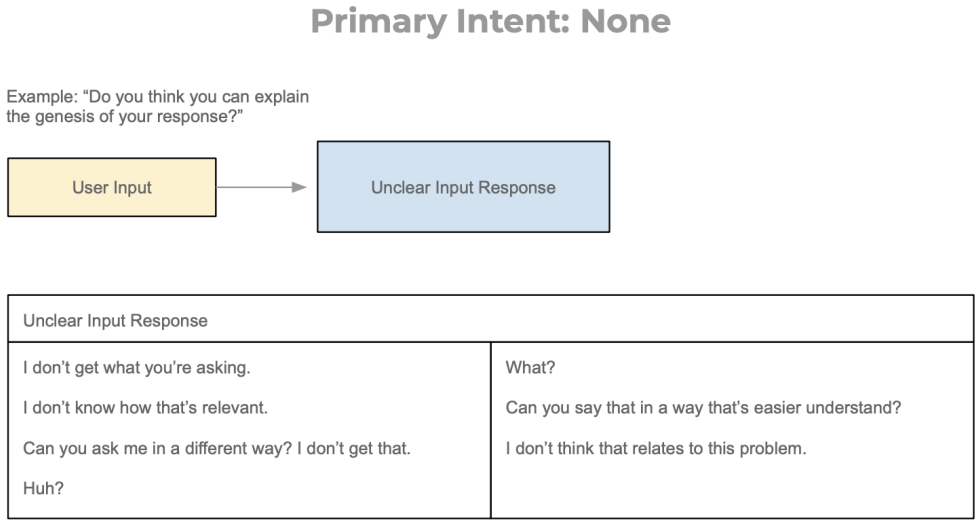



**Fig 5.13.** Logic Diagram for User Input Classified as "None"

## 5.4 Intent Classification Development and Validation

### 5.4.1 Summary

With the logic of the intent categories established in the previous section, I now move to develop a classifier to allow the logic to identify which primary and subintent match a user

input without using explicit rules or pattern matching techniques.

Conversational agents moving away from rules-based design and towards natural language processing architectures require intent classification. Tradition classification training requires large labeled data sets to have acceptable accuracy results. Within the Education domain, relevant labeled data is often not readily available, and the resources to develop and train large datasets can often be infeasible. I demonstrate a use case of utilizing transfer learning to develop multiple intent classification models for the purposes of developing a conversational agent student that could be used to practice teaching skills. I fine-tune a small labeled dataset ranging between n=45 to n=93 per class in order to fine-tune pre-trained transformer models on a total dataset of between n=212 to n=817 samples. I compare results after fine-tuning with several readily available models for classification in natural language processing tasks: BERT [17], DistilBERT[50], RoBERTa[36], AlBERT[31], and XLM[12]. In my experiment, I maintain the same hyper-parameters and report average performance evaluations over 25 trials where data is stratified and shuffled to maintain proportions where class imbalances may be present. With a dataset of n=223 and three classes, I achieved an average balanced accuracy of 90.04%. With the full dataset of n=817 with five classes, I achieved an average accuracy of 94.01%, demonstrating the ability to develop successful classification models with small labeled datasets.

As artificial intelligence(AI) methodologies advance, there is a parallel need to continue introducing AI methodologies in meaningful and practical ways across domains. Interactive learning environments such as Conversational Agents(CA) within the classroom are examples of such a practical application; however, CAs can be difficult to develop. CA design is trending away from resource-intensive, rigid rule-based methodologies in favor of dialogue policies centered on natural language processing(NLP) architectures. A primary component of CA NLP architectures is intent classification modeling. Developing new classifier models requires large labeled datasets. Within the education domain, relevant labeled data for specific use-cases is often not readily available, and the resources in terms of time, money, and even

expertise required to develop and train large datasets can be infeasible. Large pre-trained transformer models are easily accessible and can be leveraged to fine-tune small datasets via Sequential Transfer Learning(STL) for NLP classification purposes. I demonstrate the ease of utilizing STL to develop multiple intent classification models for the purposes of developing a CA within the education domain. I use a dataset of n=817 to fine-tune a primary classifier with five classes as well as subsets of the data to train three additional classifiers ranging from five to four classes with sample sizes between n=212 to n=224. I compare results between multiple transformers for sequence classification in NLP tasks: BERT [17], DistilBERT[50], RoBERTa[36], AlBERT[31], and XLM-RoBERTa[12]. I achieve an average balanced accuracy ranging from 90.94% (n=224, three classes) to 94.01% (n=817, five classes) for the highest scoring transformer, demonstrating the ability to develop successful classification models in low data and low resource settings that can be used to support more accessible NLP-based CAs with a use-case of my classifiers implemented in such a system within the education domain.

### 5.4.2 Approach

AI applications for Education are vast. One such application is purposed to assist learners by interacting with a system-supported learning environment (IBM Watson). Also, to assist teachers in providing feedback to individual learners in settings with large numbers of students or providing realistic computer-generated dialogue in teacher simulation.

Developing meaningful connections with AI in the education domain can be difficult due to insufficient data where relevant labeled data is challenging to find or not available. Also, depending on implementation, data may require a high degree of specificity, which is resource intensive to create and may not be feasible due to lack of expertise available, cost, or time required to generate.

This section addresses a way to help bridge the connection between AI methodologies and meaningful implementation within the education domain. I use a small dataset and demonstrate a solution for a common use case in AI and NLP as well as simulation technologies

that could support many developments within the education domain: intent classification in dialogue-based systems. I propose utilizing a subset of transfer learning, sequential transfer learning specific for NLP purposes, to fine-tune transformer models. I demonstrate the process of developing several intent classifiers, a key component in NLP-based conversational agent development. This component is essential in moving away from rule-based conversational dialogue systems and towards natural dialogue within a self-generated conversational system. My implementation of these classifiers is part of a larger effort to develop the ACTS system.

### 5.4.3 Experiment

I developed a dataset of n=817 samples of user input texts based on previous run-throughs of using the conversational agent, as well as attempting to ensure each class had a minimum of n=40 samples. The dataset consists of text representing user inputs for my dialogue system, each labeled with a primary label of intent and, as applicable, a sub-label as well. The composition of the developed dataset is in Table 5.2.

In my experiment, I maintain the same hyper-parameters and report average performance evaluations over 25 trials where each trial dataset is stratified and shuffled to maintain proportions where class imbalances may be present and split into a ratio with a test size of 20%. I use the dataset to fine-tune four classifiers: the "Main" classifier for all data (n=817) and three sublabel classifiers. The sublabel classifiers do not share any data points and are intended for implementation as a sequential intent as part of the CA teaching simulation dialogue policy. The three sublabel classifiers are: "Pump" classifier (n=224), "Feedback" classifier (n=212), and "Inform" classifier (n=223).

I utilize several transformers for sequential classification: BERT [17], DistilBERT[50], RoBERTa[36], AlBERT[31], and XLM-RoBERTa[12]. My hyperparameters are constant across each run. I use an AdamW optimizer, a learning rate of 2e-5, a batch size of 16, a weight decay of 0.01, and I train over ten epochs. I train using Google Colab NVIDIA Tesla-P100 GPU implemented with Huggingface[56] transformer library and training pipelines.

**Table 5.2.** Intent Classification Data

| Label | Sublabel | Samples (N) | Percentage |
|---|---|---|---|
| connect | connect | 65 | 7.96% |
| *connect Total* | | 65 | 7.96% |
| feedback | negative | 66 | 8.08% |
| | neutral | 68 | 8.32% |
| | positive | 78 | 9.55% |
| *feedback Total* | | 212 | 25.95% |
| inform | conceptual | 53 | 6.49% |
| | context | 62 | 7.59% |
| | replacement | 63 | 7.71% |
| | value | 45 | 5.51% |
| *inform Total* | | 223 | 27.29% |
| none | none | 93 | 11.38% |
| *none Total* | | 93 | 11.38% |
| pump | clarification | 67 | 8.20% |
| | conceptual | 80 | 9.79% |
| | value | 77 | 9.42% |
| *pump Total* | | 224 | 27.42% |
| **Grand Total** | | **817** | **100.00%** |

### 5.4.4 Results

**Table 5.3.** Main Intent Detailed Experiment Results

| Classifier : Main Dataset | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Model | Precision | | | | Recall | | | | F1 Score | | | | Balanced Accuracy | | | |
| | Mean | Min | Max | Std Dev | Mean | Min | Max | Std Dev | Mean | Min | Max | Std Dev | Mean | Min | Max | Std Dev |
| **RoBERTa** | **0.961**.8 | 0.932 | 0.982 | 0.014 | **0.959** .8 | 0.927 | 0.982 | 0.015 | **0.953**.8 | 0.928 | 0.982 | 0.015 | **0.940**.8 | 0.889 | 0.975 | 0.025 |
| AlBERT | 0.951 | 0.932 | 0.976 | 0.011 | 0.950 | 0.933 | 0.976 | 0.011 | 0.950 | 0.931 | 0.976 | 0.011 | 0.933 | 0.894 | 0.965 | 0.017 |
| XLM RoBERTa | 0.950 | 0.925 | 0.982 | 0.017 | 0.948 | 0.915 | 0.982 | 0.018 | 0.948 | 0.915 | 0.982 | 0.018 | 0.929 | 0.874 | 0.969 | 0.022 |
| DistilBERT | 0.949 | 0.925 | 0.982 | 0.013 | 0.948 | 0.921 | 0.982 | 0.014 | 0.947 | 0.918 | 0.982 | 0.014 | 0.924 | 0.874 | 0.976 | 0.027 |
| BERT | 0.947 | 0.922 | 0.988 | 0.016 | 0.945 | 0.921 | 0.988 | 0.016 | 0.945 | 0.921 | 0.988 | 0.016 | 0.923 | 0.878 | 0.980 | 0.023 |

The results between the transformers is comparable and in the fine-tuning process they relatively quickly are able to achieve high accuracy. This is shown in Figure 5.14. By epoch 5 all models are able to achieve a greater than 0.90 balanced accuracy average over the 25 runs. The training loss also depicts the comparability of the systems and is depicted in Figure 5.15.

I provide additional details for the fine-tuning results for the "Main" classifier in Table 5.3. I report mean, minimum, maximum, and standard deviation for precision, recall, F1 scores, and the balanced accuracy. Balanced accuracy is selected to account for the imbalance of

**Table 5.4.** Highest Average Balanced Accuracy Score over 25 Runs

| Classifier | n | Classes | Transformer | Accuracy |
|---|---|---|---|---|
| Main | 817 | 5 | RoBERTa | 94.01% |
| Pump | 224 | 3 | RoBERTa | 91.12% |
| Inform | 223 | 4 | RoBERTa | 90.94% |
| Feedback | 212 | 3 | RoBERTa | 92.84% |

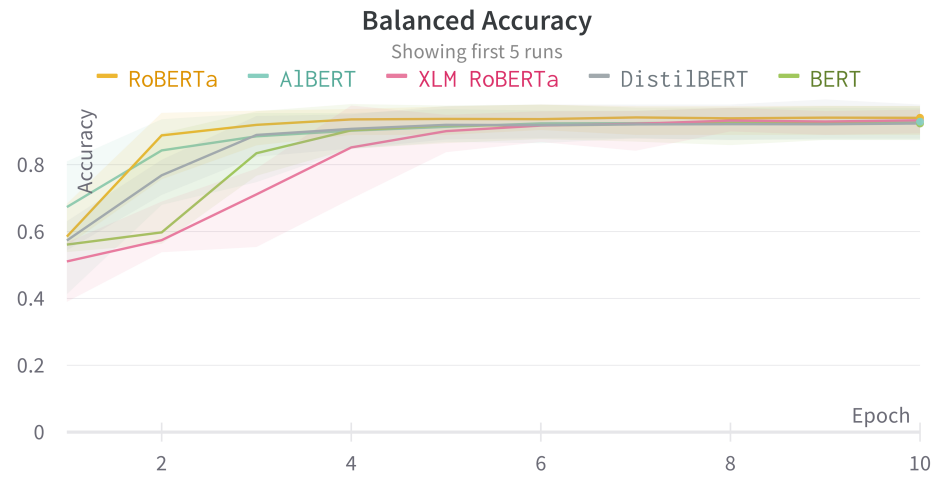



**Fig 5.14.** Balanced Accuracy by Epoch Separated by Transformer Model

classes present in the "Main" classifier dataset, where class size ranges from n=65 to n=22.

The performance in training indicates a speed advantage of certain base transformer models. The results are shown in Figure 5.16. DistilBert performs the fastest and XLM RoBERTa performs the slowest.

Additionally, Figure 5.17 reports the final confusion matrix values for the three implemented classifiers based on the full dataset with the intent classifier fine-tuned on the RoBERTa transformer model. The inform classifier was not implemented in this iteration due to the increased dialogue management complexity.

The resulting average balanced accuracy is shown in Table 5.4. These results seem high for classifiers used in educational contexts. This may be due to skewed data generation where data generated with labels in mind bias the variation found within the data, thereby causing classes to be more similar within classes to each other compared to real-world collected data. This concern is addressed further in future works. Another reason may be due to

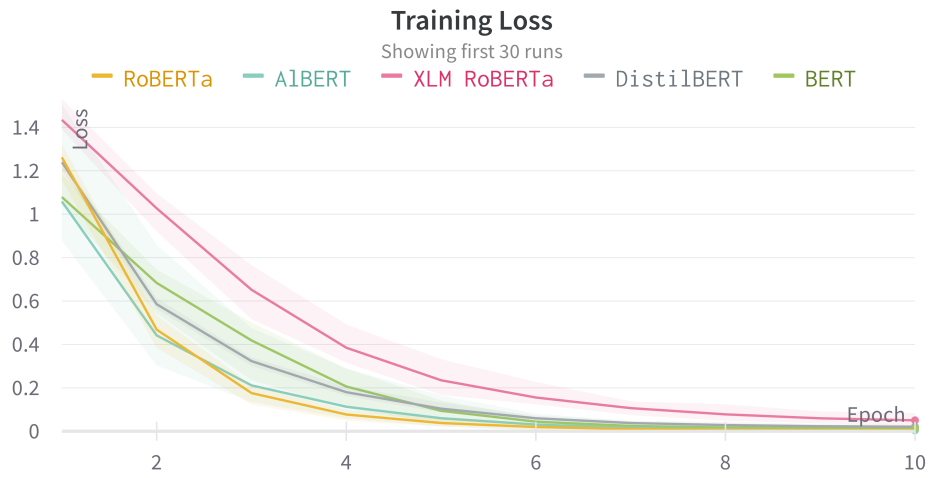



**Fig 5.15.** Training Loss by transformer Model in Fine-tuning Process

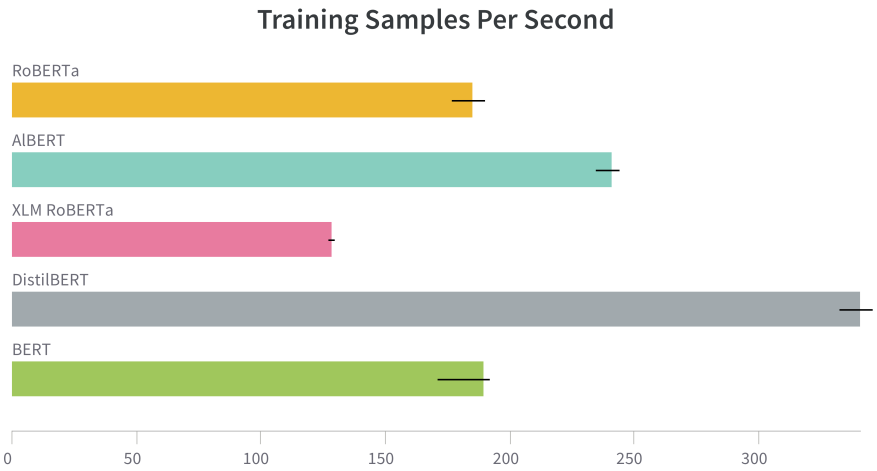
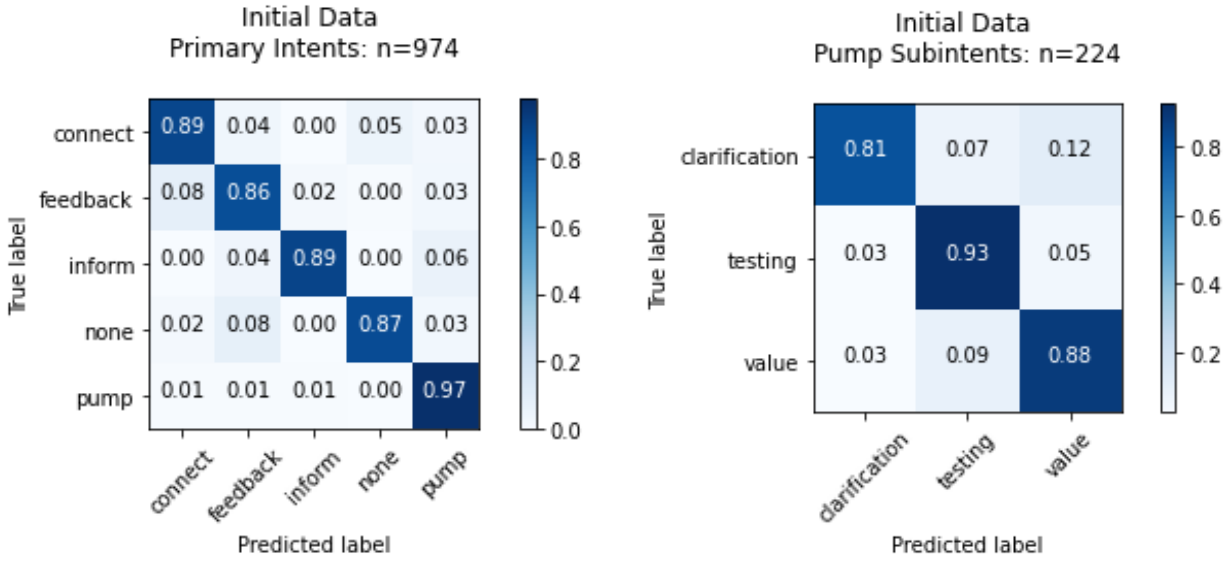



**Fig 5.16.** Training Samples per Second by Transformer Model

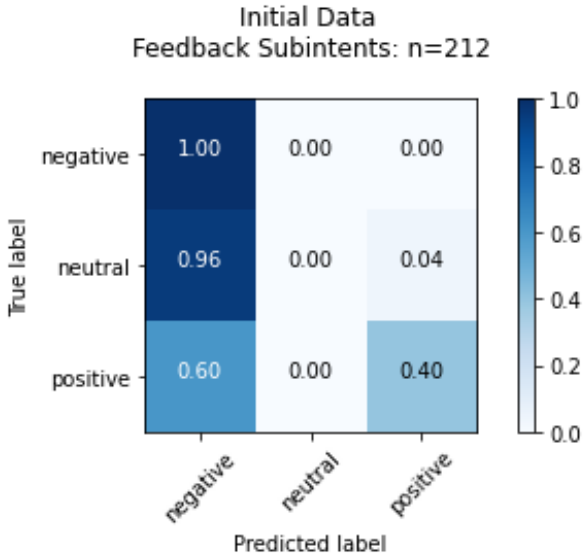
the scope of the implementation context being constrained to the task intended for this specific CA development. However, this is also great news for those planning to do task-based classification (e.g., in teaching simulations), where the concern has been that narrowly defined contexts will not have data sets large enough for accurate classification with NLP, I can achieve high accuracies with insufficient data given a narrow scope of implementation.

I demonstrate a viable option for sequence classification for NLP purposes by using Sequence Transfer Learning with Transformer models. Results show several transformers that can all be implemented based on scenario requirements. For example, the fine-tuned




**(a)** Main Intent Classifier

**(b)** Pump Subintent Classifier




**(c)** Feedback Subintent Classifier

**Fig 5.17.** Confusion Matrix Results: RoBERTa Transformer Classifier Model by Intent Category

XLM-RoBERTa classifier performs nearly as well as other models and could be used in the case of planning for future multilingual implementations of a CA) I provide a method to support bridging the connection between AI advances, as they specifically relate to CA development and the education domain such as for improved generated responses in CAs for teaching simulations.

A fundamental limitation of this intent classification modeling is that real-world implementation is likely to see lower classification accuracies because the data was constructed artificially. This represents the unknown unknowns as I develop a system and categories for intents.

## 5.5 Conclusion

I propose a framework for developing entities with consideration of generalizable structures. Additionally, I provide a framework for intent classification categories with the corresponding logic for an initial simplistic iteration of a dialogue management system. I conclude the chapter by providing information from a preliminary experiment on intent classification accuracy. In these discussions, I have furthered the contributions from Chapter 3 by providing a case study for implementing the proposed process of conversational agent development. To summarize my contributions, I frame the resulting impacts in terms of the proposed research questions at the beginning of the chapter:

> **Research Question 5.1**: *With insights from preliminary testing, what changes can I implement to improve the design of a pedagogical teachable agent? Can these improvements be demonstrated to allow for transparency in the development of a conversational agent process?* I provide transparency into the development process and illustrate with a case study iteration the use of my proposed process. I identify a crucial insight regarding meta-process agents requiring separation from intent classification and feedback mechanism implementation.

**Research Question 5.2**: *In a no data, minimal time scenario, what is an effective way to deploy a new intent classification component within a conversational agent design?* With minimal resources, I develop a replacement intent classification structure demonstrating a transparent process of no-data, high accuracy, and quick-turn classification. I utilize industry methods to further research development, emphasizing a critical insight of simplification in using an agile-based development framework.

**Research Question 5.3**: *When building a dialogue management system, how can I utilize a generalizable structure to minimize the requirements in developing additional scenarios?* I propose a novel coding framework in pedagogical conversational agents that incorporates dynamic variable structures in design discussions. My implementation in the code allows for scenario scalability in future development efforts and provides a generalizable implementation insight to improve development best practices within the field.

# Chapter 6

# System Deployment, User Study, and Recommendations

## 6.1 Motivation

I conclude my efforts by completing the process of conducting a novel study where I demonstrate the completion of the proposed process and provide a discussion of the evaluation metrics from Chapter 3.

The purpose of my user study is not only to provide an example demonstration and thereby complete the first iteration of the implementation of the development process proposed in Chapter 3 but also to demonstrate a critical element in conversational agent evaluation, establishing a gold standard to compare with and de-conflate varying metric categories as discussed in Chapter 3.

I then further contribute to the discussion with a review of the results, which provide insight into pedagogical system development, and finally, I focus on the evaluation components. The research questions I address are as follows:

> **Research Question 6.1**: With the difficulty associated with comparing niche conversational agents with each other, can I demonstrate establishing a baseline

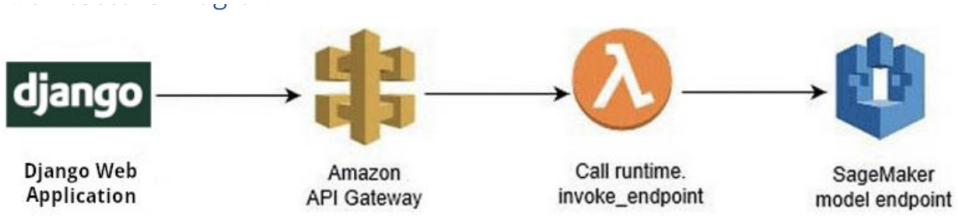



**Fig 6.1.** SageMaker Endpoint Implementation Architecture

and a gold standard in the conversational agent development process? *This addresses a gap in the literature for Conversational Agent Gap 3 in Section 2.1.6.*

**Research Question 6.2**: What insights can be gained from completing a real-world test of the system? *This addresses gaps in the literature for Conversational Agent Gap 3 in Section 2.1.6.*

**Research Question 6.3**: How do the proposed evaluation metrics compare with previously identified conversational agent metrics in literature? *This addresses gaps in the literature for Niche Gap 2 in Section 2.3.*

## 6.2 Deployment of System

I deployed the system on Amazon Web Services (AWS) to allow for system scaling.

There are two primary components in the deployment process. The first is to offload classifier models to Amazon Web Services so that the computational effort required of local servers is minimal. This is necessary to minimize the lag time of the response generation of a system when user inputs need classification. Creating a SageMaker endpoint requires several services linked to allow a developer to reference the model and utilize the classifier. The general architecture connecting the code is shown in Figure 6.1.

The second component of the system's deployment is instantiating an EC2 instance and creating a link where users can access the system from any computer. This is discussed in Section 6.2.5. The following sections discuss the deployment specifics of the SageMaker endpoint, followed by an elaboration on the EC2 implementation.

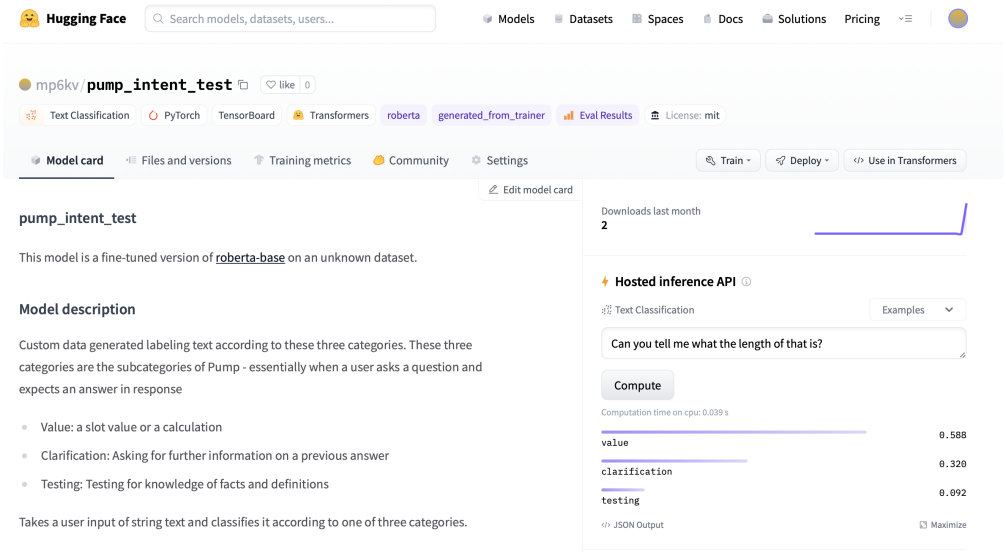



**Fig 6.2.** Huggingface Classifier Model Deployment

### 6.2.1 Huggingface Model Deployment

I used the platform Huggingface (shown in Figure 6.2) to train and deploy the machine learning models required for this project.

### 6.2.2 Amazon Web Services: SageMaker

Once the models have been built and trained in Huggingface, I use the SageMaker HuggingFace Inference Toolkit to deploy those models to Amazon Web Services: SageMaker. This is shown in Figure 6.3. These models are deployed through a Jupyter Notebook and managed in a SageMaker Studio Domain. This deployment iteration results in four models hosted on SageMaker: a paraphrase model, the primary or "main" intent model, a pump intent model, and a feedback model, all discussed in detail in the previous chapter.

### 6.2.3 Amazon Web Services: Lambda

As shown in Figure 6.4, AWS Lambda is an event-driven, serverless computing service. I have four different SageMaker endpoints running, each with its own Lambda function to interface with. Each Lambda function has identical code but different endpoint names, allowing multiple configurations. For a very high-level look, when an event is received, the Lambda function logs it to AWS CloudWatch and transforms it into a Python object that

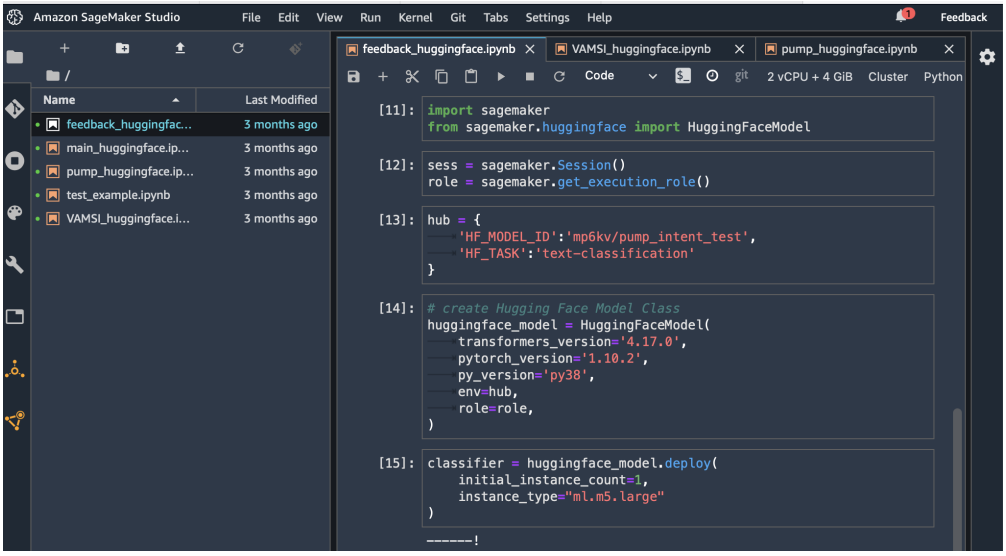



**Fig 6.3.** Amazon Webservices Deployment: SageMaker Studio Example Code

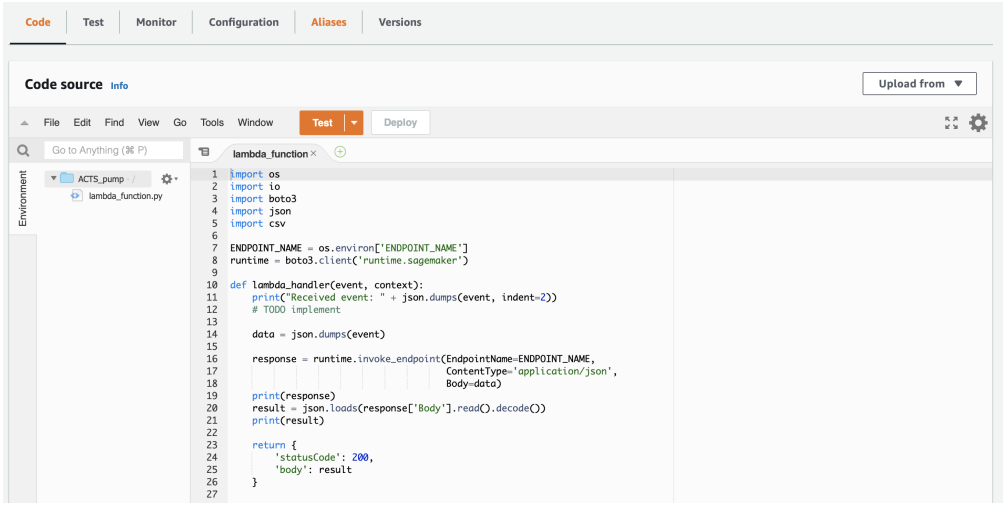



**Fig 6.4.** Amazon Webservices Deployment: Lambda Example Code

can be sent to SageMaker.

### 6.2.4 Amazon Web Services: API Gateway

These Lambda functions require an API to interface with. I use Amazon Web Services: API Gateway, shown in Figure 6.5. This API has four different resources handling POST requests, each resource interacting with one Lambda function.

Figure 6.7 displays the API gateway connection to SageMaker completing the link. The classifier model hosted on SageMaker can now be accessed with a single line of code without needing much processing power. When the SageMaker implementation was first deployed,

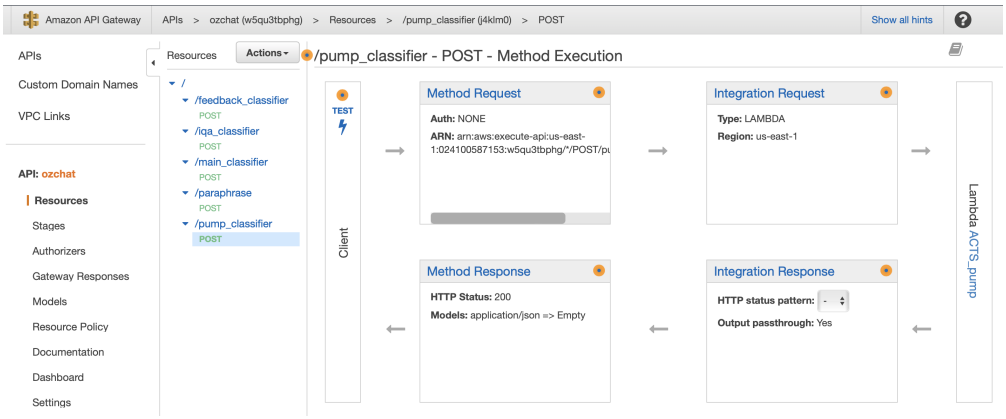



**Fig 6.5.** Amazon Webservices Deployment: API Gateway

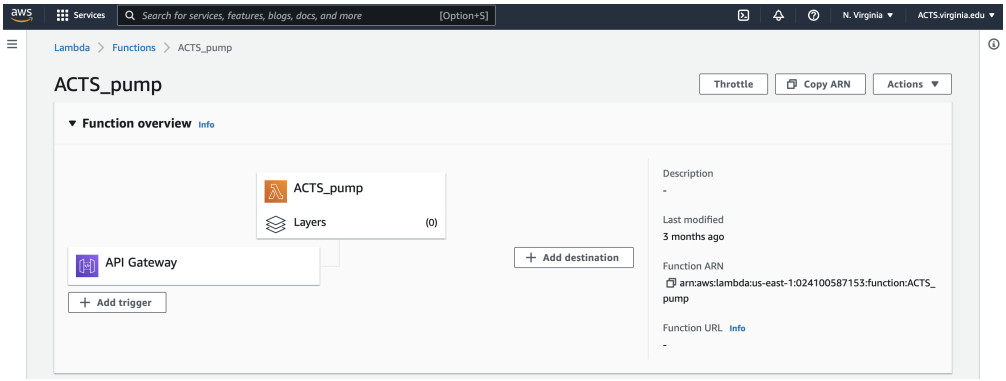



**Fig 6.6.** Amazon Webservices Deployment: Lambda Connection to API Gateway and SageMaker

the latency of computer responses was shortened from 15-20 seconds to 2-5 seconds.

### 6.2.5 Amazon Web Services: System Deployment EC2 Instance and Elasticache

Within the structure of the code, I utilize REDIS to save all conversational data within a JSON structure and load it between turns so that if the session was disconnected, the information would be retained and could be referenced again. This architecture within the code is shown in Figure 6.7.

To implement a REDIS structure on AWS, I implemented an Elasticache service to host the REDIS datastore. The EC2 Architecture implemented is shown in Figure 6.8 where I continue to use a REDIS store by utilizing the AWS Elasticache service, and I use Nginx as a webserver.

When creating the EC2 instance, I have provided runtime instructions within the code

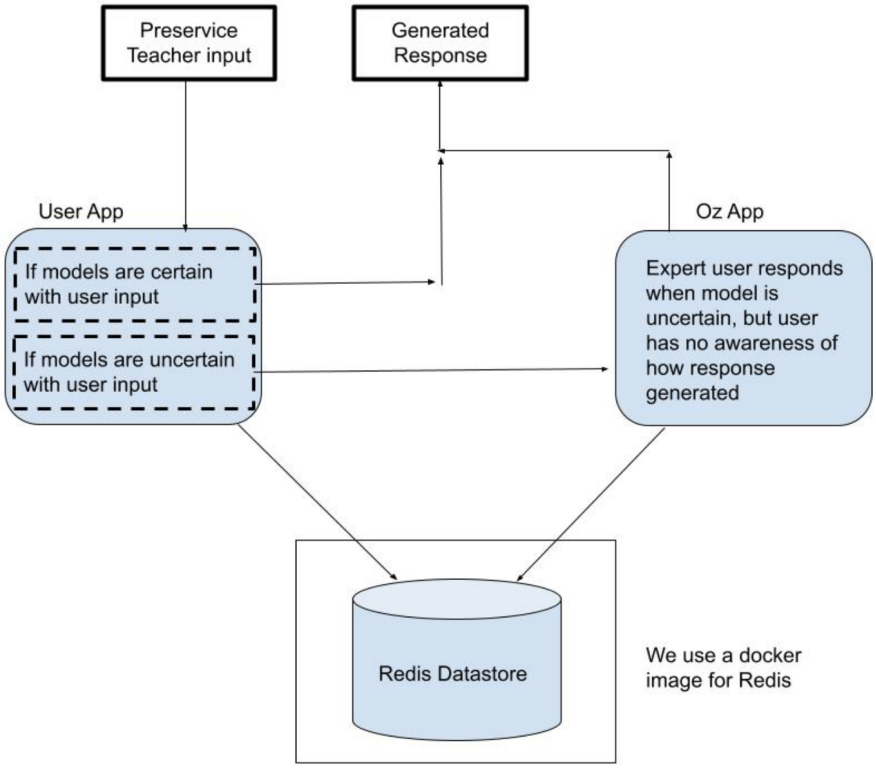



**Fig 6.7.** REDIS Configuration Architecture

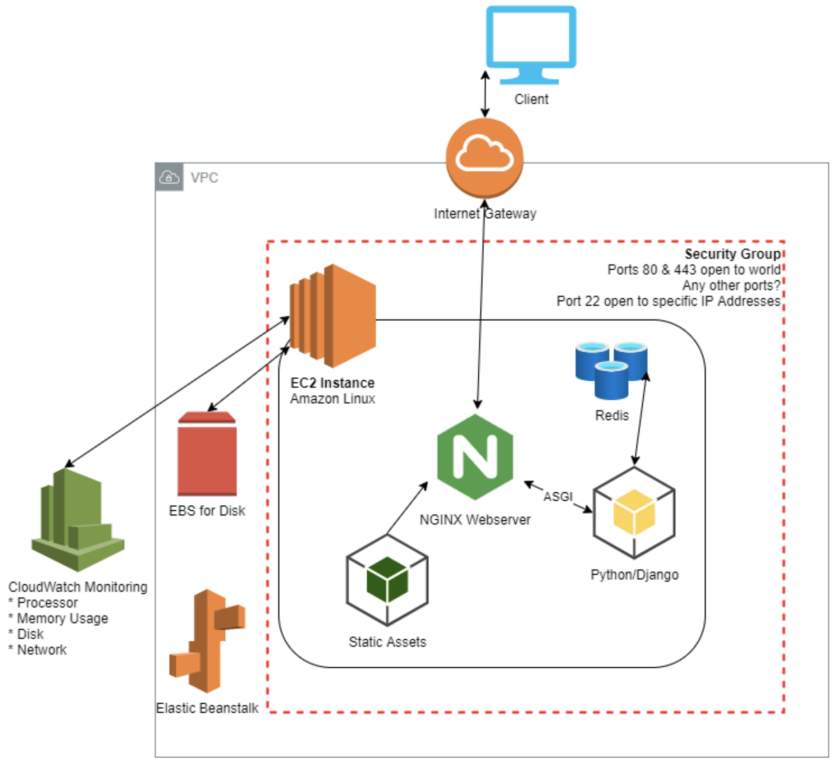



**Fig 6.8.** Amazon Webservices Deployment Architecture

that instruct a user how to SSH into the instance and connect to the Github repository where the code is stored. The Github repository includes an initialization file that, when activated, will execute all needed actions to initiate the instance correctly. This file incorporates installing dependencies on a new EC2 instance.

## 6.3 Methodology

### 6.3.1 Participants

The study was done from May 2022-July 2022. A total of 19 participants ranging from ages 26-71 and 2-35 years of teaching experience, located in Virginia, California, Florida, Georgia, Colorado, and South Carolina completed the study. The study consisted of an online demographic and consent form survey followed by a one-hour video Zoom call where participants interacted with the ACTS system for as many sessions as time allowed. Eleven participants taught STEM subjects while eight taught Humanities subjects; twelve participants identified as Female and seven identified as Male. Participants taught grades from Kindergarten through college. Inclusion criteria for participants are that they were actively teaching or had taught in the United States, speak English fluently, are United States citizens, understand scale factor at an eighth-grade level, had access to video Zoom call, and had stable internet for the duration of the session.

### 6.3.2 Configuration Explanations

I conducted an experiment using Design of Experiments (DoE) with two configurations. The configurations are:

- Gold Standard: Facilitator acts as Oz and responds for each turn as if they were the virtual student
- Standard Baseline: Facilitator engages when the conversational agent is not confident or does not meet set thresholds for intent classifications

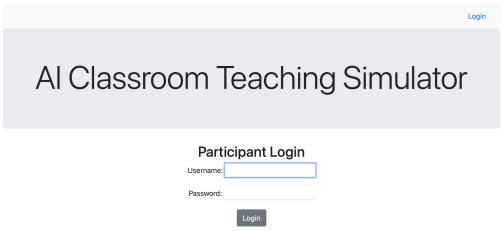



**Fig 6.9.** Conversational Agent Session Example: Login

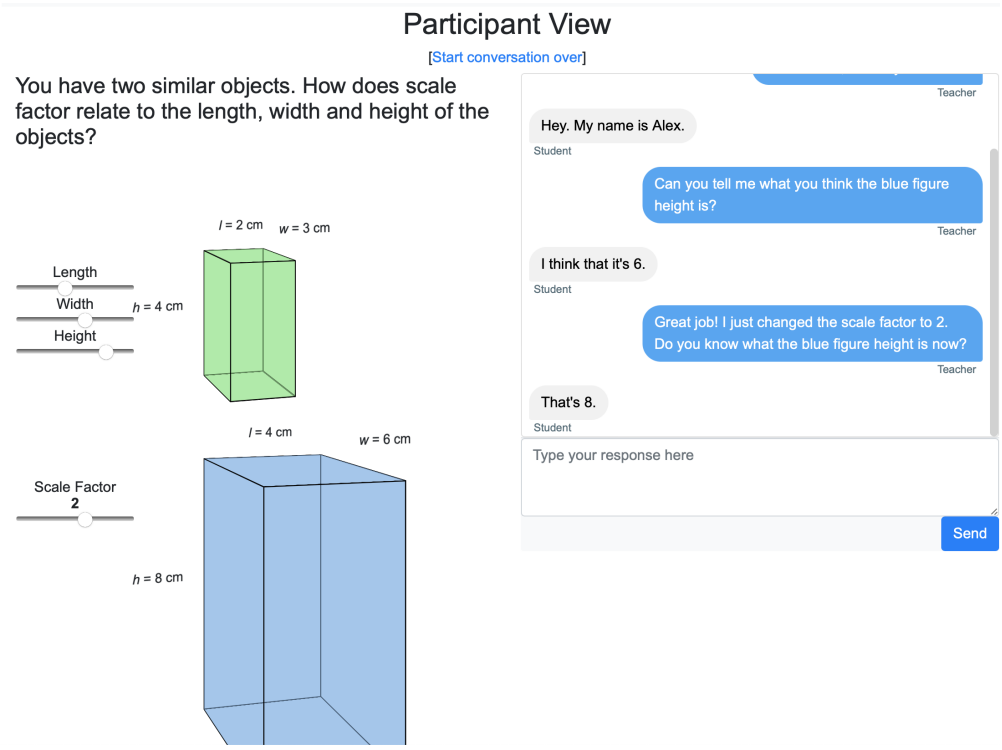



**Fig 6.10.** Conversational Agent In-Session Example

### 6.3.3 System Interaction Example

Each participant was assigned a random number and first engaged the ACTS system with the BS or GS configuration. The average number of completed sessions was 1.79 sessions. The Gold Standard Ozchat protocol imitated a student who believed that the scale factor was added to a dimension to achieve the updated dimension value. The student was asked a question, "You have two similar objects. How does scale factor relate to the length, width, and height of the objects?" The participant was prepped with the idea that they would initiate the conversation and try to help the student respond to the question successfully.

Figures 6.9 and 6.10 show the user view of the system and example dialogue by the

system. Participants had access to a sliding scale to change the values of the object's length, width, and height, as well as the scale factor. The virtual student did not have access to the image sliders, and they were not able to see the blue figure values.

When the participant felt ready to end the conversation, if time was short within the study, or if there was an error with the system, the conversation would end. After each interaction, participants completed a study detailing their views of the realism of the conversational agent interaction. The questions used are the same as those in Table 3.3.

## 6.4 Results

I collected text data, transcripts, surveys, as well as metadata and analyzed the results. I noticed a significant distinction between the user perception of the Baseline Standard(BS) version and the gold standard version. Figure 6.11 depicts the categories of the survey questions with each boxplot pair containing the two filters by configuration type. The colored box indicates the space between the second and third quartiles, with the color changing inside the box at the median value. The "whiskers" extending out indicate the maximum and minimum values. In this boxplot, the x-axis holds two values: the Baseline Standard (the average of survey responses after interacting with the conversational agent) and the Gold Standard (the average of survey responses after interacting with the facilitator as the conversational agent). The six pairs of data on the x-axis can be considered six separate boxplots, all sharing a y-axis. "Realism" questions were split out into Student Realism (how realistic the user perceived the student's interactions to be) and Scenario Realism (how realistic the user perceived the overall scenario and interface to be). The y-axis is the average value of survey responses on a scale of 1-5. Some survey questions were inverted (i.e., higher values indicated less sensible), and these values were appropriately inverted to match the rest of the data. In all six categories, users consistently rated the Gold Standard variation higher on all metrics for the main body of responses (the colored-in section, which contains the second to third quartile of data). However, maximum values for the Baseline Standard consistently reached up to 5, and minimum values for the Gold Standard consistently reached

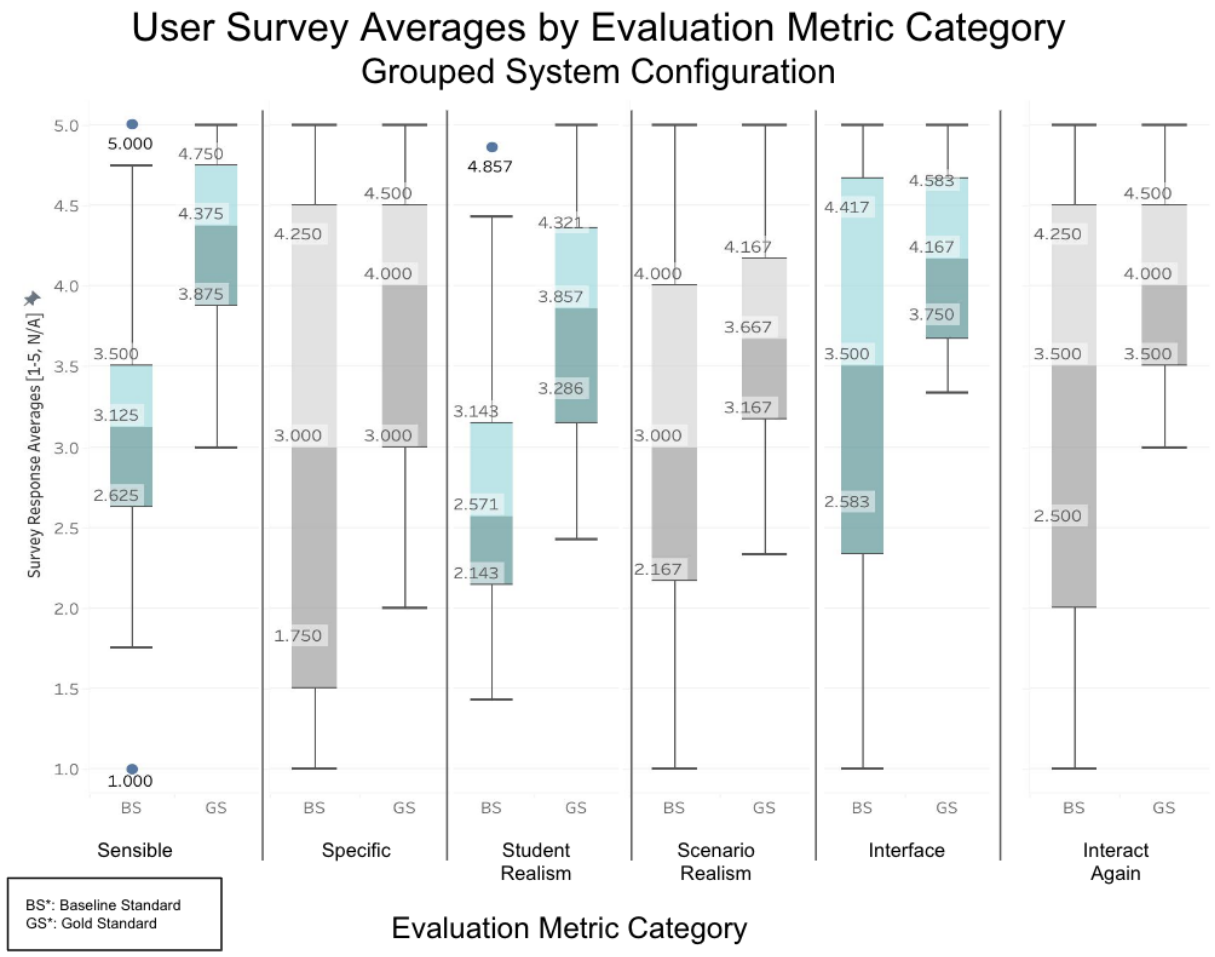



**Fig 6.11.** Box Plots

down to the median of the Baseline Standard. In other words, some users consistently rated the conversational agent with a perfect score, and some users consistently rated a human facilitator as less human than the conversational agent! This highlights that even with a gold standard of human interaction, limitations in the interface and user expectation of student understanding still affected the results. However, comparing the Baseline Standard and Gold Standard allows for separating out the impacts of this user bias and interface design. As an additional note not reflected in the data, a significant number of users desired more context to the scenario, citing that in their experience as teachers, they were accustomed to relying on posture, tone of voice, and other factors not present in this scenario.

While inspecting the data, I noticed another variable that seemed to have a noticeable

impact on survey ratings - prior familiarity with scale factor. The results are visible in Figure 6.12. In this boxplot, the x-axis holds sets of two values, reflecting the user's familiarity with scale factor. The colored box indicates the space between the second and third quartiles, with the color changing inside the box at the median value. The "whiskers" extending out indicate the maximum and minimum values. All users were expected to have basic competence on the subject (and were given a review sheet), so the only two options were "Unsure/Other" to indicate limited or tentative comfort with scale factor concepts or "Yes" to indicate complete comfort with scale factor concepts. The six pairs of data on the x-axis can be considered six separate boxplots, all sharing a y-axis. "Realism" questions were split out into Student Realism (how realistic the user perceived the student's interactions to be) and Scenario Realism (how realistic the user perceived the overall scenario and interface to be). The y-axis is the average value of survey responses on a scale of 1-5. Some survey questions were inverted (i.e., higher values indicated less sensible), and these values were appropriately inverted to match the rest of the data. Interestingly, in all but two categories (Student Realism and Interface), users who were more familiar with the topic of scale factor were more likely to rate the conversational agent higher. Users who were less familiar with scale factor were more critical of the conversational agent, yet simultaneously less critical of the interface and more willing to interact with the student again. Given the limited sample size, some of these distinctions may not be statistically significant.

Finally, I conclude with a validation of the intent classification model proposed in Chapter 5. I now test the model on real-world data and rate the accuracy. The accuracy of the data developed in an austere environment was 0.94 using the RoBERTa transformer model throughout 25 randomized runs with a test sample size of 20%. I collected 843 user inputs in this real-world study and ran the intent classification models with that data. I then labeled the ground truth for each data point and compared the results. The resulting confusion matrix is in Figure 6.13. The resulting accuracy is 0.80. One noticeable feature in this confusion matrix is that there is only one ground truth value for the "none" category, which

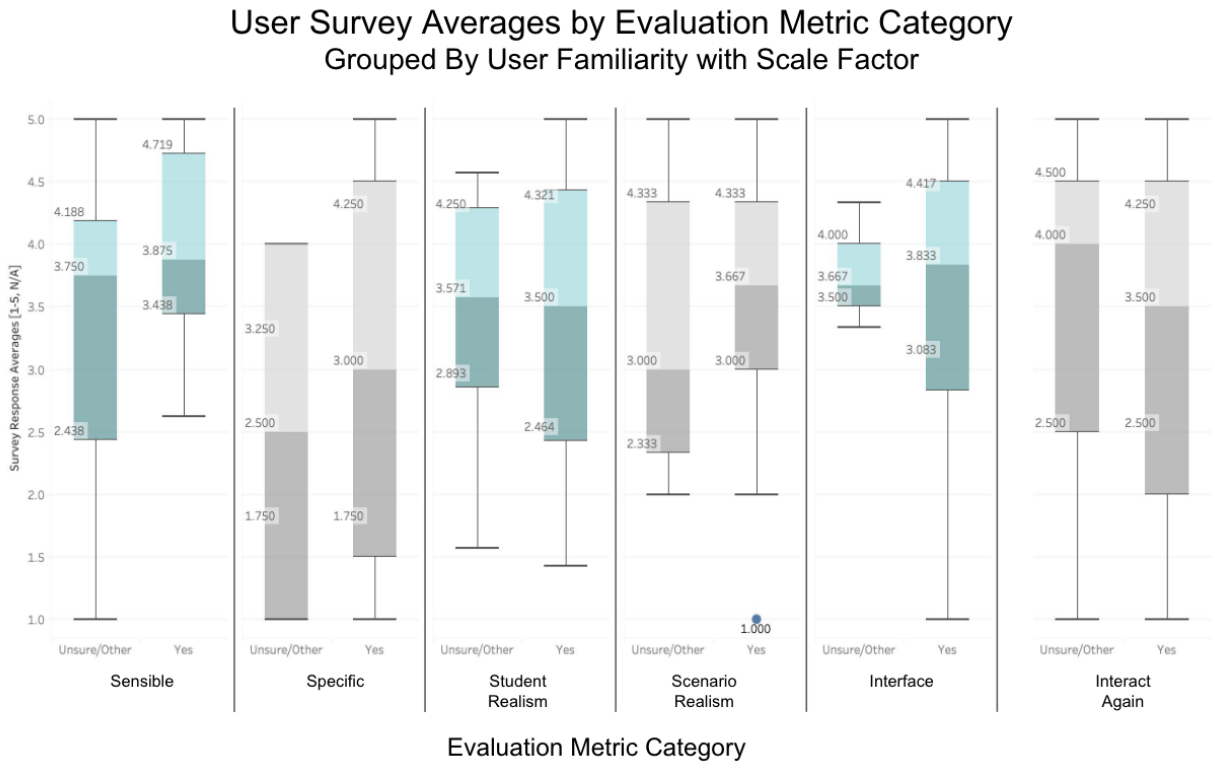



**Fig 6.12.** Box Plot of User Survey Averages by Evaluation Metric Category Grouped by Familiarity with Scale Factor

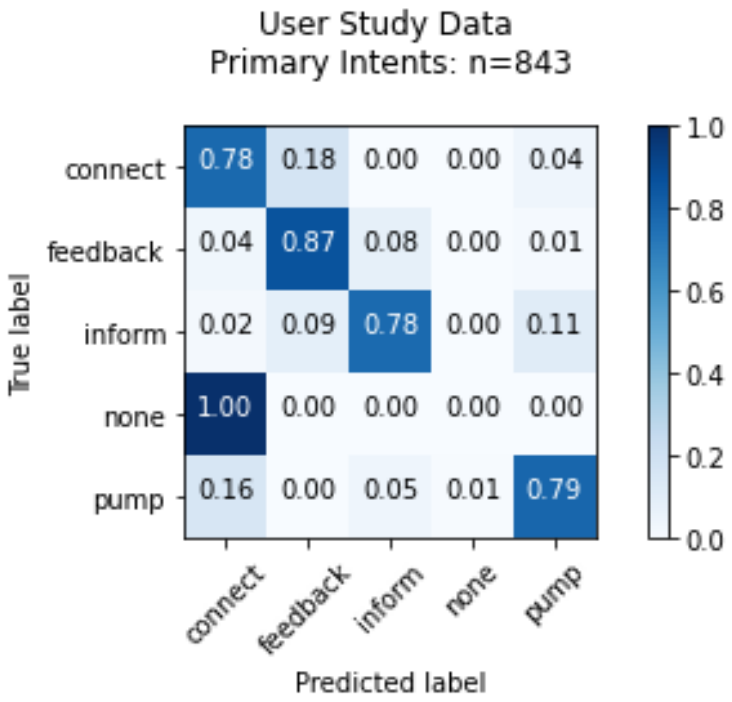
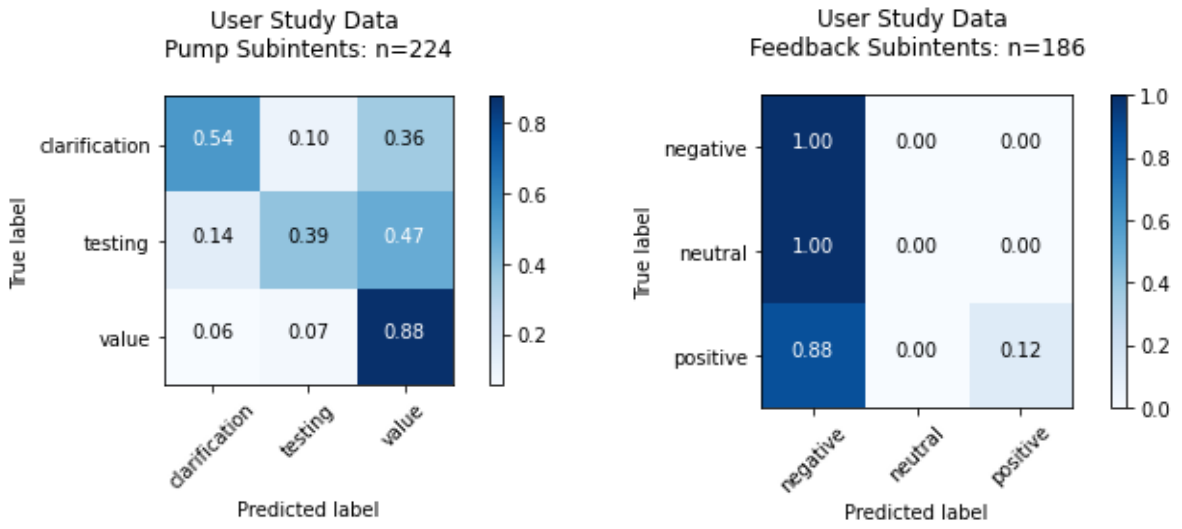



**Fig 6.13.** Confusion Matrix of Main Intent Categories

the model misclassified. This leads us to believe this category is not needed as participants tend to stay on track with their conversations in a predictable way, and for those that do not stay on track, I can utilize the confidence threshold for model predictions.

Figure 6.14 shows the confusion matrix results for subintent classifiers for "Pump" and "Feedback". The feedback classifier performs abysmally in classification, demonstrating a bias toward negative feedback. No neutral feedback labels were identified. This indicates poor categorization to begin with or errors in training the classifier model, which requires additional exploration to fine-tune an appropriate model for future iterations of the system.

Table 6.1 offers a breakdown of the time spent in various configurations. The tone is set with Average Total Turns, as users took an average of 1.37 times as many turns interacting with a real human in the Gold Standard variation as they did with the conversational agent in the Baseline Standard variation (note that the numbers in parentheses include instances where technical errors cut the session artificially short). It is worth remembering, though, that in this case, efficiency is not necessarily a priority or even a goal at all - the conversational




**(a)** Primary Intent: Pump **(b)** Primary Intent: Feedback

**Fig 6.14.** User Study Confusion Matrix Results for Subintent Classifiers

**Table 6.1.** User Study Results Using Traditional Quantitative Metrics

| Configuration | Average Total Turns | Average Time | Average time per turn |
|---|---|---|---|
| **BS** | 11.82 (9.3) | 13 min 46s | 1 min 18s |
| **GS** | 16.15 | 18 min 14s | 1 min 8s |
| **Total** | 14.61 (13.15) | 15 min 27s | 1 min 4s |

agent could start the interaction already understanding the problem if efficiency was the goal. Since a realistic interaction with a realistically confused student is the goal instead, a lower number of turns taken to finish the interaction is not self-evidently better. The greater number of turns taken with the Gold Standard version is reflected in the average time of the interactions, with the Gold Standard version again taking approximately 1.3 times as long as the Baseline Standard variation. This confirms the trend that users spend longer interacting with the Gold Standard variation. Interestingly, however, the average time per turn column indicates that while they take longer overall, users interact faster with the Gold Standard variation, taking more turns but completing them faster. The difference is slight, however, and could be explained by factors other than user response time (for instance, the facilitator might be able to send a reply marginally quicker than the conversational agent).

Table 6.2 shows the frequency with which the facilitator intervened on behalf of the conversational agent in a Baseline Standard variation. Gold Standard variations are not

**Table 6.2.** User Study Results Using Relevant Quantitative Metrics

| Measure | Intervention by Oz |
|---|---|
| **Average per session** | 3.1 |
| **Percent intervention by Oz** | 30.10% |

included since they are by default exclusively the facilitator interacting on behalf of the conversational agent. Roughly a third of the time, when reading user input, the system could not produce a reasonable output. While it could undoubtedly be higher, that is still a substantial proportion of the time. In future iterations of this conversational agent architecture, lowering that percentage could be another measure of improvement.

## 6.5 Discussion

In addition to survey results, I collected notes and asked additional questions of the patients as time allowed. I asked participants what would make a more realistic student, had they encountered a teachable agent prior to this experience, and do they see a benefit to the purpose of this system. I additionally allowed for comments to be provided that they may not have had the chance to provide throughout the session. This led to an ability to conduct a thematic analysis, and below are some of the themes that consistently presented themselves:

- Participant perceived particular conversational agent and scenario qualities oppositely. For example, some participants were enthusiastic that the virtual student paraphrased their inputs, while other participants prematurely ended the session after receiving a paraphrased response multiple times in a row
- 100% of participants had never encountered a "teachable agent."
- 95% of participants saw a clear benefit for this type of tool being incorporated into preservice teaching programs.
- 47% of participants commented on the difficulty of capturing varying student types and capabilities when asked about what could improve the realism of the student.

- >50% of participants did not have a structured response for what could improve the realism of students, although the majority did not give full marks to the realism of the system regardless of which configuration they interacted with

Additionally, some of the observations I noticed are:

- Preference/Dislike of a TA that repeated back what they heard
- Technology limitations within implementation categories - some users were unable to connect to the internet or use a computer as proficiently as other participants, and they were less likely to engage with system features such as slider bars
- Handful of participants felt very limited by the lack of being able to engage in a more relationship-building interaction-way - stated they would usually joke with the student
- Limitations of system interface had a noticeable impact - Participants desired to send multiple messages rather than be a turn-by-turn interaction. Additionally, over half of the participants commented on using the "Enter" button rather than click-to-send messaging.

Overall there is a distinction between the versions of the system and a rich amount of data to be explored as further relationships within the data emerge.

## 6.6 Proposed System Alterations

Several issues in deployment of the system were encountered and tracked. These issues may have been a logic error in the coding or a system issue that needs to be addressed. For example, the paraphraser transformer model implemented cut off generated paraphrase outputs and needs to be addressed in order to operate as intended. Aside from these code and implementation level errors, there are several larger system iteration improvements and alterations recommended. Several improvements and alterations are discussed in detail in the subsequent subsections.

### 6.6.1 Improved Intent Classification

To continue the development process, developers need to track progress and improvements after a user study or real-world test. Based on the information from the intent classification categories, I can alter the intent classifications to match the true nature of conversation flow better when using the system. For example, as opposed to Table 5.1, I can now alter the categories to eliminate unnecessary sections and further elaborate or section other intents to better assist with logic flow. I do so in Table 6.3.

In order to implement this suggestion, the design and dialogue management and flow will need to be reimagined and planned for and, in step 2 of the development process, re-coded to align to these categories.

### 6.6.2 Improved Feedback Mechanism and Meta-purpose Agent Goal integration

The modified Instructional Quality Assessment (IQA) feedback mechanism needs to be further developed to achieve the desired purpose of the conversational agent. Currently there is only a classifier that is developed. The most accurate classifier needs to be deployed to SageMaker so it can best be incorporated in deployment of the system. The capability is currently limited to summing the total count of each user utterance. In order to provide meaningful feedback, developers will need to work with domain experts to better understand what meaningful feedback is and then attempt to model and recreate that feedback within the system. One idea to move forward with this topic is to classify each utterance from the user study using the IQA classifier, attempt to identify any patterns, and review conversations line by line with domain experts while capturing examples of feedback the domain experts would provide given the IQA classification and context of the conversation. Key to this integration is starting with foundational and achievable attempts using a similar methodology to the AGILE process. While keeping the result in mind, attempt to establish foundational and baseline versions that can be used as initial attempts even if they do not encompass the full desired result of the feedback mechanism.

**Table 6.3.** Intent Category Descriptions Recommendations Post Real World Test and Evaluation of System

| Main Intent | Sub Intent | Description | Example |
|---|---|---|---|
| Connect | | Beginning an interaction or building a connection with a student external to direct education goal | *Hi, how are you today?* |
| Pump | Value | Asking the student to provide some value response: numerical output | *Can you tell me what the length is for the green box?* |
| | Conceptual | Testing student understanding of a topic related to definitions, theoretical, and equations | *Can you tell me what a reduction is?* |
| | Context | Asking about problem displayed or what the student is struggling with | *What do you notice about the figures?* |
| | Clarification | Asking for the student to provide further information to a previous response. | *Which object did you mean is bigger?* |
| Inform | Value | Providing a value to the student | *The scale factor is now 2.* |
| | Conceptual | Providing theoretical, definition, or equation information | *The equation for the volume is length times width times height.* |
| | Context | Providing information about the context of the problem | *We are focusing on the blue object volume.* |
| | Replacement | Replacing information the student previously understood as true | *The value of the green length is actually 6.* |
| Feedback | Positive Feedback | Encouragement that the student is correct with no suggestions | *Yes, that's right. Good Job.* |
| | Neutral Feedback | Acknowledging a student utterance with no indication of direction. | *Ok, I hear you.* |
| | Negative Feedback | Indicating that the student's understanding is not correct or the student needs to change direction. | *I see where you're going with that, but that's not exactly the full picture.* |

**Table 6.4.** Evaluation of Virtual Student Responses Framework with Example Responses to question: "How does surface area relate to scale factor?"

| **Category** | **Characteristic** | **Response Example** | **Teacher Perspective Example** |
|---|---|---|---|
| Non Normative | There is not enough context or information provided to assess the correctness | It's cubed. | What does the student think is cubed? |
| Partially Normative | The answer may be correct but the context is unclear. More information is required to assess student's understanding of the concept. | Squared. | What does the student think is squared? |
| Normative | Aligns with expected response and demonstrates accuracy | The scale factor is squared. | There is enough precision to assess that the student understands the concept. |

### 6.6.3 Establish Virtual Student Response Metric

In order to improve realism, establishing a relevant metric and developing a classifier for evaluation purposes is desired. I recommend further development of a comparison using the Table 6.4 as an initial point during the next development iteration. With this framework, resulting responses can be labeled and measured for realism.

### 6.6.4 Anthropomorphic Virtual Student Quality Inclusion: Fallibility and Growth in Understanding Scenario Topic

Vital in representation of a real student is the ability to misunderstand a topic and the ability to grow in understanding of a topic. Some elements of growth in learning are represented in this iteration. For example, a user may provide updated information of the true dimension values of the system.

There are many more elements that can be expanded upon, however, such as incorporating a trigger that will allow the virtual student to access pre-programmed vital knowledge if the users input is semantically similar to the content (e.g. reminding the virtual student of a correct equation for volume of a 3-D object).

I also recommend developing different levels of understanding and perception of the

student. Nascent efforts were incorporated via structure within the code, but they are not active in the current deployment. This code provides a suggested structure to allow for varied virtual student levels of understanding. For example a "beginner" student may add scale factor to obtain a new dimension value rather than multiply. An "advanced" student may correctly calculate scale factor and the volume and surface area of an object the majority of the time but still occasionally have a mathematical error. This effort to capture the fallibility of student understanding is vital to the realism component of the system.

I recommend first pursuing simplistic mathematical and functional differences between the varied understanding levels. A future development to build off of this foundational work could attempt to address the nuances of conceptual differences between student understanding. This is difficult to do because the elements of developing a response to conceptual questions can be hard-programmed and explicitly written out, but the main thrust of the ACTS effort is to develop minimally rules-based hard-programmed responses. The majority of hard-programmed responses center on not knowing information rather than provided content specific answers. For example it is the difference between responding "I don't understand, can you ask that in a different way?" and hard-coding a response of "The scale factor gets multiplied by the dimensions so that must mean it's cubed to get the volume".

The varied amount of question types and responses is large and it is not recommended to pursue a complex decision-tree approach in order to accomplish this hard-coded but perhaps more realistic response to a finite and minimal amount of questions. To pursue a hard-coded path explicitly or outright deters from state-of-the-art generative methods and would regress in novelty from a systems engineering perspective.

### 6.6.5 Future Experiment Design

One final recommendation is careful crafting of future experiment designs. If attempting to evaluate the entire system then here are several recommendations to incorporate in study design and development:

- Conduct user studies early and often in design construction

- Incorporate Design of Experiments in experiment construction
- Minimize total configurations
- Consider timing in data collection with annual school-schedule for pedagogical agents
- Research methods to recruit eligible participants such as intentional or perhaps avoidance of social media outreach [29]
- Consider intentional evaluation construction

If attempting individual components such as evaluating a metric of normative-ness as discussed in Section 6.6.3, then consider evaluation through less intensive efforts than a system-wide user study. One such example may be identifying common or key scenario questions, establishing a normative response, then evaluating the generated response of the conversational agent through domain-expert labeling. For this example see Figure 6.15 where I identified fifteen relevant questions with corresponding normative responses. Several questions have several semantically similarly worded phrasing that could be randomly implemented on a given iteration of the system. These questions could be asked of the response-generation component of the system and captured, imported to a labeling website where domain experts could then assess the perception of normative given the response. This could provide developers insight into whether a generative response can achieve normative-ness and if so, what are ways or versions of the system that can capture different ranges of normative-ness. The insights provided from such an evaluation could then be used in designing improved logic components for varied student understanding levels as discussed in Section 6.6.4.

This example is one such implementation in design studies for future works although expansions can vary and these suggestions are not all-encompassing of recommended work.

1. **Can you tell me what scale factor is?**
    a. How do you define scale factor?
    b. What is a scale factor?
    c. What do you know about scale factor?
- A number that multiplies the dimensions of a shape.
- The ratio of the length of a side of one figure to the length of the corresponding side of the other figure.
- The ratio between corresponding measurements of an object and a representation of that object.
- A number that multiplies times a given quantity to produce a smaller or larger version of the original number

2. **What is the equation for scale factor?**
    a. What is the formula to calculate scale factor?
    b. How would you calculate scale factor?
- y=Cx where C is the scale factor
- To find the scale factor from a smaller object to a larger object you can divide the larger figure measurement by the smaller figure measurement
- To find the scale factor from a larger object to a smaller object you can divide the smaller figure measurement by the larger figure measurement
- You calculate the scale factor of similar figures by taking the ratio of corresponding parts of the two figures. When enlarging the shape, the larger measurement is the numerator, and the smaller measurement is the denominator. When shrinking the shape, the smaller measurement is the numerator, and the larger measurement is the denominator.

3. **How does volume relate to scale factor?**
    a. How does scale factor change the volume of an object?
- The volume is multiplied by the scale factor cubed

4. **How does area relate to scale factor?**
    a. How does scale factor change the surface area of an object?
- The area is multiplied by the scale factor squared

5. **How does perimeter relate to scale factor?**
    a. How does scale factor change the perimeter of an object?
- The perimeter is multiplied by the scale factor

6. **How does scale factor change the length of an object?**
- The length is multiplied by the scale factor

7. **How does scale factor change the width of an object?**
- The Width is multiplied by the scale factor

8. **How does scale factor change the height of an object?**
- The Height is multiplied by the scale factor

9. **Can you explain what happens to scale factor when an object is enlarged?**
- The scale factor is greater than one.

10. **Can you explain what enlargement is?**
- It's also called scaling or dilation and can make an object bigger or smaller according to a scale factor.

11. **What happens to an object when the scale factor is less than 1?**
    a. What happens to an object when the scale factor is smaller than 1?
- The object gets smaller

12. **What happens to an object when the scale factor is greater than 1?**
    a. What happens to an object when the scale factor is bigger than 1?
- The object gets larger

13. **What happens to an object when the scale factor is equal or the same as 1?**
- The object is the same size

14. **Does scale factor always make things bigger?**
- No it can also make things smaller

15. **Do the dimensions (length, width, height) of an object always all change by the same amount when scale factor is applied?**
- No, while they are multiplied by the same they aren't all added the same

**Fig 6.15.** Example Evaluation Questions with Crafted Normative Responses

## 6.7 Conclusion

With the completion of the study, I have empowered the ACTS team to identify further areas of development needed to improve the ACTS system. I demonstrate the implementation of the novel process and evaluations discussed in Section 3. I utilize mixed methods and the proposed evaluation approach and validate results. Finally, in addition to capturing system recommendations, I capture real-world data to validate the intent classification model I developed and also validate categories that are intuitive and helpful for conversational agent design when designed for a teachable agent role. I address the research questions and provide closing remarks for each listed here:

**Research Question 6.1**: *With the difficulty associated with comparing niche conversational agents with each other, can I demonstrate establishing a baseline and a gold standard in the conversational agent development process?* I demonstrate the value of establishing a gold standard to disaggregate measures within conversational agent evaluation and provide an example of how to construct and

implement a design of experiments study to establish a gold standard for comparison during development.

**Research Question 6.2**: *What insights can be gained from completing a real-world test of the system?* I completed a study on the improved version of the ACTS system and completed the proposed process by conducting a thorough evaluation and recommending improvements to the system. I validate the need for alternative evaluation metrics for pedagogical teachable agents and empower researchers to further expand on my work.

**Research Question 6.3**: *How do the proposed evaluation metrics compare with previously identified conversational agent metrics in literature?* I validate the need for alternative evaluation metrics for pedagogical teachable agents and empower researchers to further expand on my work.

I continue on to Chapter 7, in which I summarize the contributions of this overall effort, the limitations, and the future works recommended.

# Chapter 7

# Discussion and Conclusion

## 7.1 Discussion

Let us sum up what I have covered so far. I began by investigating, in broad strokes, the current state of literature for pedagogical, skill-oriented student conversational agents. And I found that there are still significant gaps in the research in this area.

First, there are too few such specialized conversational agents, and those that exist are far behind the most modern innovations in natural language processing. With several possible use cases (foremost of which is the opportunity for pre-service educators to hone their skills), this field is worthy of investigation and innovation. Second, I also note that there is a still-developing understanding of methods to generate intent classifiers in a low- or no-data starting environment. With limited structured data for such a specific or "niche" domain, there is an opportunity to develop an innovative method for generating the initial classifiers in an austere environment. And third, there is an ongoing debate in the literature about evaluating conversational agents and measuring and quantifying their success or failure.

Furthermore, most of the already limited research in this area suggests evaluation metrics that would fit a conversational agent used in an industry-based application, where accuracy and efficiency are paramount. These goals, however, would actively set back a teachable conversational agent - to provide realistic opportunities for pre-service teacher experience,

the conversational agent would pointedly need not to have the correct or most efficient answers. The entire point of the agent is to allow the user to guide the agent to those answers.

Seeking to move the established knowledge base forward on all three of these counts, I designed and deployed the initial prototype of a virtual student pedagogical conversational agent with the support of my team. The next area of proposed contribution is developing the preliminary models without access to domain-specific structured data. With a modified version of IQA classification, I successfully trained the prototype's models using classroom transcripts separated to the sentence level. This method is a state-of-the-art technique for iteratively developing domain-specific conversational agents without the vast amounts of data usually required for such an endeavor.

After the first tests of the prototype, I set out to iterate the prototype and perform a real-world test. In this, I reach toward the first area of innovation mentioned above. This is the first conversational agent of its kind - a retrieval-generative teachable agent designed to test and hone the user's skill rather than support student learning of the content of the conversation topic. The ACTS teachable agent is a novel development, and the study performed upon it is the first to investigate the effects and opportunities of such an interaction.

Finally, while investigating the data from the above study, I proposed and applied a novel set of metrics to evaluate conversational agents, with allowance made for pedagogical and other realism-centric agents. In a context lacking consensus, I consider situations in which accuracy and consistency are not the highest priorities, but instead a realistic encounter that affords the user an opportunity to develop pedagogical skills as they guide and instruct the teachable agent. With that in mind, I propose a novel set of evaluation metrics predicated on the "Ozchat" framework to assess the teachable agent's behavior relative to an actual human.

I believe that my teachable agent contributes new knowledge to the body of literature on

all three counts mentioned. By its essence, it is a new kind of conversational agent not created or tested before. I pioneer methods to train models without domain-specific structured data in a novel codified process. And in my evaluation of its strengths and weaknesses, as revealed in my user study, I propose modified metrics and methodologies by which to develop and evaluate pedagogical conversational agents.

## 7.2 Limitations

The ACTS teachable agent system provides the first of its kind attempt at a natural language dialogue system for pedagogical teachable agents. I do not attempt to develop the most advanced natural language dialogue system that performs the most sensibly or realistically. Instead, I take a practical approach in attempting to develop a system in a no-to-low-data scenario that can be replicated and is transparent for others in their research. I provide discussion and propose strategies that address a generalizable architecture; I provide evaluation metrics that consider the unique nature of pedagogical teachable agents. I establish a framework to continue to develop conversational agents in an iterative process. From the niche within the domain to the contribution to the literature, I provide novel discussion and results to the conversation of pedagogical teachable agents.

Some of the limitations of our results include that this is a limited domain scoped conversational agent. I provide code to replicate the logic I utilized for the system; however, as there are many gaps within natural language processing, natural language understanding, and natural language generation, I have not surpassed the difficulties in that field.

## 7.3 Future Works

Future works should seek to iteratively improve components of the conversational agent design while further codifying the process and validating metrics. The critical component to develop from the perspective of the success for the ACTS is an enhanced feedback mechanism utilizing the already developed modified Instructor Quality Assessment (IQA) feedback classifier. Creating a more meaningful feedback mechanism will not only allow for a more

robust fulfillment of the purpose of this teachable agent, but it will also provide a novel contribution to the literature on teachable agents focused on skills work.

There are many additional future works and ways to expand this effort. Some elements already exist in the system or were discussed in detail in Chapter 6. The list here is meant to include a wider range and summarize directions for development. From a conversational agent design perspective, suggested future works are listed in Table 7.1.

Additionally, future work can expand on components that are specific to niche fields within Natural Language Processing(NLP). A recommendation of future works specific to incorporating NLP techniques and state-of-the-art is provided in Table 7.2. These elements can be isolated and individually developed in parallel to the development of the ACTS. To contribute to the literature not only in the NLP subfield, they can use the baseline system of ACTS, incorporate the elements developed, and assess the impact on the overall conversational agent and session conversation. This overarching implementation and assessment is needed within the fields of conversational agent development specifically with pedagogical agent design and meta-purpose agent design.

## 7.4 Conclusion

I conclude by providing a significant novel contribution to the literature in conversational agents, pedagogical agents, and more specifically pedagogical teachable agents. With contributions and support from my research team, a prototype of ACTS was developed with insight and details of the process. I extended the teams initial contribution by codifying a development process, establishing processes for evaluation and development of critical conversational agent components in low-to-no data scenarios. I iterated one cycle using my proposed detailed AGILE development process to include utilizing my proposed processes in intent classifier development, dialogue management logic development, system coding, and evaluation capture of the system. I conducted and completed a user study that established the "gold standard" for ACTS as well as established a baseline to be evaluated against in future iterations of the system. I provide meaningful insights to extend my work and list

**Table 7.1.** Proposed Directions for Future Works and Improvements: Systems Engineering Design Perspective

| **Field or Category** | **Subfield or Method** | **Description** |
|---|---|---|
| Conversational Agent Design | Dialogue Management System | Improve logic flow for dialogue management elements given specific intent classifications |
| Conversational Agent Design | Feedback Mechanism | Incorporate modified IQA classification and work with domain-experts to develop meaningful feedback that can be provided in-session or post-session. |
| Conversational Agent Design | Anthropomorphic quality: Fallibility | Expand on work to further allow conversational agent to represent varied student understanding levels such as "beginner", "intermediate" and "advanced" |
| Conversational Agent Design | Anthropomorphic quality: Learning | Expand on work to further allow conversational agent to grow in concept understanding during session. |
| Conversational Agent Design | Evaluation | Meta-purpose agent evaluation considerations including expanding on proposed process for developing appropriate metrics. |
| Conversational Agent Design | Evaluation | Pedagogical teachable agent, further develop metrics that are applicable to improve virtual student realism. |
| Interface and System | User Feature Integration | Incorporate key elements that increase ease of use and realism of scenario from interface perspective such as: Enter Key, Speech-to-Text, Text-to-Speech. |
| Scenario-level | New Scenario Deployment | Attempt demonstrating system scalability to additional scenarios to encourage further generalizable design incorporation. |

**Table 7.2.** Proposed Directions for Future Works and Improvements: Natural Language Processing Methods

| **Field or Category** | **Subfield or Method** | **Description** |
|---|---|---|
| Natural Language Processing | Intent Classification | Iterate on initial intent classification categories taking into consideration proposed improved categories. |
| Natural Language Processing | Co-reference Resolution | Find and replace all references referring to the same entity in a given user input sequence to allow for improved computer understanding on. logic resolution in the dialogue management of the system. |
| Natural Language Processing | Numeracy | Improve ability for user input to be appropriately responded to by identifying and correctly tagging numbers in user input. |
| Natural Language Processing | Topic Modeling/Keywords | Useful in assisting dialogue flow. Can be used to assist inferring context of user inputs. |
| Natural Language Processing | Named Entity Recognition | Incorporate advanced Named Entity Recognition to assist in conversation context tracking and intent classification logic |
| Natural Language Processing | Text Simplification | Breaking compound sentences into smaller or more simple sentences to allow for better computer understanding of input sentences to include co-referencing development. |
| Natural Language Processing | Text Generation | Improve or replace models and rules-based hard-coded portions of code that paraphrase or respond in specifically coded test phrases. |

directions for future works from a systems engineering perspective. I additionally provide detailed paths forward for development of certain components within the dialogue system such as a proposed next iteration of the intent classifier categories. These contributions establish a contribution within the literature in the nascent development of truly natural language conversational agent developments.

# List of Publications


[1] Phillips, M., Chiu, J. L., Watson, G. S., Brown D. E. Pedagogical Conversational Agent Evaluation and Codified Conversational Agent Development Process Framework In Low Data Scenario [In Preparation]

[2] Phillips, M., Chiu, J. L., Watson, G. S., Brown D. E. Dialogue Management System for Pedagogical Agents [In Preparation]

[3] Phillips, M. (2021, June). Leveraging Unstructured Text Within the Context of Conversational Agents. In Proceedings of the 29th ACM Conference on User Modeling, Adaptation and Personalization (pp. 308-314).

[4] Datta, D.(*First Author), Phillips, M.(*First Author), Bywater, J. P., Chiu, J., Watson, G. S., Barnes, L., Brown, D. (2021, April). Virtual pre-service teacher assessment and feedback via conversational agents. In Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications (pp. 185-198).

[5] Datta, D., Phillips, M., Bywater, J. P., Lilly, S., Chiu, J., Watson, G. S., Brown, D. E. (2022). Human-in-the-Loop Data Collection and Evaluation for Improving Mathematical Conversations. In M. M. Rodrigo, N. Matsuda, A. I. Cristea, V. Dimitrova (Eds.), Artificial Intelligence in Education. Posters and Late Breaking Results, Workshops and Tutorials, Industry and Innovation Tracks, Practitioners' and Doctoral Consortium (pp. 551–554). Springer International Publishing.

[6] Mostafavi, M., Phillips, M., Jiang, Y., Porter, M. D., & Freedman, P. (2021), "A tale

of two metrics: Polling and financial contributions as a measure of performance." In 2021 IEEE International Systems Conference (SysCon) (pp. 1-6). IEEE.

# References


[1] The Importance, Natura and Impact of Teacher Wuestions. volume Proceedings of the twenty-sixth annual meeting, North American Chapter of the International Group for the Psychology of Mathematics Education, Toronto, 2004. Ontario Instittute for Studies in Education of the University of Toronto. OCLC: 61520135.

[2] Deborah Loewenberg Ball. With an Eye on the Mathematical Horizon: Dilemmas of Teaching Elementary School Mathematics. *The Elementary School Journal*, 93(4):373–397, 1993. Publisher: University of Chicago Press.

[3] Tim Bettridge. A Bot that can Talk about Anything:, August 2020.

[4] Melissa Boston. Assessing Instructional Quality in Mathematics. *The Elementary School Journal*, 113(1):76–104, September 2012. Publisher: The University of Chicago Press.

[5] Melissa Boston, Jonathan Bostic, Kristin Lesseig, and Milan Sherman. A Comparison of Mathematics Classroom Observation Protocols. *Mathematics Teacher Educator*, 3(2):154–175, March 2015. Publisher: National Council of Teachers of Mathematics Section: Mathematics Teacher Educator.

[6] Melissa D. Boston and Amber G. Candela. The Instructional Quality Assessment as a tool for reflecting on instructional practice. *ZDM*, 50(3):427–444, June 2018.

[7] Christopher J. Buttimer, Joshua Littenberg-Tobias, and Justin Reich. Designing Online Professional Learning to Support Educators to Teach for Equity During COVID and

Black Lives Matter. *AERA Open*, 8:23328584211067789, January 2022. Publisher: SAGE Publications Inc.

[8] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. Universal Sentence Encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174, Brussels, Belgium, November 2018. Association for Computational Linguistics.

[9] Catherine C. Chase, Doris B. Chin, Marily A. Oppezzo, and Daniel L. Schwartz. Teachable Agents and the Protégé Effect: Increasing the Effort Towards Learning. *Journal of Science Education and Technology*, 18(4):334–352, August 2009.

[10] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. Reading Wikipedia to Answer Open-Domain Questions. *arXiv:1704.00051 [cs]*, April 2017. arXiv: 1704.00051.

[11] Nalin Chhibber and Edith Law. Using Conversational Agents To Support Learning By Teaching. page 7, 2019.

[12] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online, 2020. Association for Computational Linguistics.

[13] Tara Dalinger, Katherine B. Thomas, Susan Stansberry, and Ying Xiu. A mixed reality simulation offers strategic practice for pre-service teachers. *Computers & Education*, 144:103696, January 2020.

[14] Debajyoti Datta, Valentina Brashers, John Owen, Casey White, and Laura E. Barnes. A Deep Learning Methodology for Semantic Utterance Classification in Virtual Human

Dialogue Systems. In David Traum, William Swartout, Peter Khooshabeh, Stefan Kopp, Stefan Scherer, and Anton Leuski, editors, *Intelligent Virtual Agents*, Lecture Notes in Computer Science, pages 451–455, Cham, 2016. Springer International Publishing.

[15] Debajyoti Datta, Maria Phillips, James P Bywater, Jennifer Chiu, Ginger S Watson, Laura Barnes, and Donald Brown. Virtual Pre-Service Teacher Assessment and Feedback via Conversational Agents. page 14, 2021.

[16] Debajyoti Datta, Maria Phillips, James P. Bywater, Jennifer Chiu, Ginger S. Watson, Laura E. Barnes, and Donald E. Brown. Evaluation of mathematical questioning strategies using data collected through weak supervision, December 2021. arXiv:2112.00985 [cs].

[17] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[18] Sarah E Finch and Jinho D Choi. Towards Unified Dialogue System Evaluation: A Comprehensive Analysis of Current Evaluation Protocols. page 10, 2020.

[19] Shuyang Gao, Abhishek Sethi, Sanchit Agarwal, Tagyoung Chung, and Dilek Hakkani-Tur. Dialog State Tracking: A Neural Reading Comprehension Approach, August 2019. arXiv:1908.01946 [cs].

[20] Melody Guan, Varun Gulshan, Andrew Dai, and Geoffrey Hinton. Who Said What: Modeling Individual Labelers Improves Classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), April 2018. Number: 1.

[21] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching Machines to Read and Comprehend. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[22] Sebastian Hobert and R. Wolff. Say Hello to Your New Automated Tutor - A Structured Literature Review on Pedagogical Conversational Agents. In *Wirtschaftsinformatik*, 2019.

[23] Matthew B. Hoy. Alexa, Siri, Cortana, and More: An Introduction to Voice Assistants. *Medical Reference Services Quarterly*, 37(1):81–88, January 2018. Publisher: Routledge _eprint: https://doi.org/10.1080/02763869.2018.1404391.

[24] Johnny Hung. masterfung/scrapy-craigslist, December 2020. original-date: 2014-10-16T05:26:38Z.

[25] Victor Hung, Miguel Elvir, Avelino Gonzalez, and Ronald DeMara. Towards a method for evaluating naturalness in conversational dialog systems. In *2009 IEEE International Conference on Systems, Man and Cybernetics*, pages 1236–1241, October 2009. ISSN: 1062-922X.

[26] Shafquat Hussain, Omid Ameri Sianaki, and Nedal Ababneh. A Survey on Conversational Agents/Chatbots Classification and Design Techniques. In Leonard Barolli, Makoto Takizawa, Fatos Xhafa, and Tomoya Enokido, editors, *Web, Artificial Intelligence and Network Applications*, Advances in Intelligent Systems and Computing, pages 946–956, Cham, 2019. Springer International Publishing.

[27] Brian Junker, Yanna Weisberg, Lindsay Clare Matsumura, Amy Crosson, Mikyung Kim Wolf, Allison Levison, and Lauren Resnick. Overview of the Instructional Quality Assessment: (644942011-001). Technical report, American Psychological Association, 2006. type: dataset.

[28] Jihun Kim and Minho Lee. Robust Lane Detection Based On Convolutional Neural Network and Random Sample Consensus. In Chu Kiong Loo, Keem Siah Yap, Kok Wai Wong, Andrew Teoh, and Kaizhu Huang, editors, *Neural Information Processing*, Lecture Notes in Computer Science, pages 454–461, Cham, 2014. Springer International Publishing.

[29] Thomas Krueger. Finding Better Research Participants and Avoiding Fraud, October 2018.

[30] Mohammad Amin Kuhail, Nazik Alturki, Salwa Alramlawi, and Kholood Alhejori. Interacting with educational chatbots: A systematic review. *Education and Information Technologies*, July 2022.

[31] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. *arXiv:1909.11942 [cs]*, February 2020. arXiv: 1909.11942.

[32] Raina Langevin, Ross J Lordon, Thi Avrahami, Benjamin R. Cowan, Tad Hirsch, and Gary Hsieh. Heuristic Evaluation of Conversational Agents. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–15, Yokohama Japan, May 2021. ACM.

[33] Edith Law, Parastoo Baghaei Ravari, Nalin Chhibber, Dana Kulic, Stephanie Lin, Kevin D. Pantasdo, Jessy Ceha, Sangho Suh, and Nicole Dillen. Curiosity Notebook: A Platform for Learning by Teaching Conversational Agents. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI EA '20, pages 1–9, New York, NY, USA, April 2020. Association for Computing Machinery.

[34] Patrick Lewis, Pontus Stenetorp, and Sebastian Riedel. Question and Answer Test-Train Overlap in Open-Domain Question Answering Datasets. *arXiv:2008.02637 [cs]*, August 2020. arXiv: 2008.02637.

[35] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. Recurrent Neural Network for Text Classification with Multi-Task Learning, May 2016. arXiv:1605.05101 [cs].

[36] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv:1907.11692 [cs]*, July 2019. arXiv: 1907.11692.

[37] Antoine Louis. A Brief History of Natural Language Processing — Part 1, July 2020.

[38] Antoine Louis. A Brief History of Natural Language Processing — Part 2, July 2020.

[39] Noboru Matsuda, Evelyn Yarzebinski, Victoria Keiser, Rohan Raizada, William W. Cohen, Gabriel J. Stylianides, and Kenneth R. Koedinger. Cognitive anatomy of tutor learning: Lessons learned with SimStudent. *Journal of Educational Psychology*, 105(4):1152–1163, November 2013.

[40] Lindsay Clare Matsumura, Brian Junker, Yanna Weisberg, and Amy Crosson. Overview of the Instructional Quality Assessment, 2006. Place: University of California, Graduate School of Education & Information Studies Publisher: National Center for Research on Evaluation, Standards, and Student Testing (CRESST).

[41] Sanjana Mendu, Mehdi Boukhechba, Janna R. Gordon, Debajyoti Datta, Edwin Molina, Gloria Arroyo, Sara K. Proctor, Kristen J. Wells, and Laura E. Barnes. Design of a Culturally-Informed Virtual Human for Educating Hispanic Women about Cervical Cancer. In *Proceedings of the 12th EAI International Conference on Pervasive Computing Technologies for Healthcare*, pages 360–366, New York NY USA, May 2018. ACM.

[42] Erinc Merdivan, Deepika Singh, Sten Hanke, and Andreas Holzinger. Dialogue Systems for Intelligent Human Computer Interactions. *Electronic Notes in Theoretical Computer Science*, 343:57–71, May 2019.

[43] Rashmi Metri. Chatbot 101 — From the history to the future of Chatbots, April 2021.

[44] Sarah Michaels, Catherine O'Connor, and Lauren B. Resnick. Deliberative Discourse Idealized and Realized: Accountable Talk in the Classroom and in Civic Life. *Studies in Philosophy and Education*, 27(4):283–297, July 2008.

[45] Joao Luis Zeni Montenegro, Cristiano André da Costa, and Rodrigo da Rosa Righi. Survey of conversational agents in health. *Expert Systems with Applications*, 129:56–67, September 2019.

[46] Robert C. Pianta and Bridget K. Hamre. Conceptualization, Measurement, and Improvement of Classroom Processes: Standardized Observation Can Leverage Capacity. *Educational Researcher*, 38(2):109–119, March 2009. Publisher: American Educational Research Association.

[47] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ Questions for Machine Comprehension of Text, October 2016. arXiv:1606.05250 [cs].

[48] Alexander J Ratner, Christopher M De Sa, Sen Wu, Daniel Selsam, and Christopher Ré. Data Programming: Creating Large Training Sets, Quickly. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[49] Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Kurt Shuster, Eric M. Smith, Y.-Lan Boureau, and Jason Weston. Recipes for building an open-domain chatbot. *arXiv:2004.13637 [cs]*, April 2020. arXiv: 2004.13637.

[50] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv:1910.01108 [cs]*, February 2020. arXiv: 1910.01108.

[51] Nihit Saxena. Chatbot Tutorial: Choosing the Right Chatbot Architecture, February 2020. Library Catalog: towardsdatascience.com.

[52] Pavel Smutny and Petra Schreiberova. Chatbots for learning: A review of educational chatbots for the Facebook Messenger. *Computers & Education*, 151:103862, July 2020.

[53] Zhuoran Wang and Oliver Lemon. A simple and generic belief tracking mechanism for the Dialogue State Tracking Challenge: On the believability of observed information. *Proceedings of SIGDIAL 2013*, August 2013. Publisher: Association for Computational Linguistics.

[54] Joseph Weizenbaum. ELIZA- a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1):36–45, January 1966.

[55] Jason D Williams. Web-style ranking and SLU combination for dialog state tracking. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 282–291, Philadelphia, PA, U.S.A., 2014. Association for Computational Linguistics.

[56] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *arXiv:1910.03771 [cs]*, July 2020. arXiv: 1910.03771.

[57] Zhao Yan, Nan Duan, Peng Chen, Ming Zhou, Jianshe Zhou, and Zhoujun Li. Building Task-Oriented Dialogue Systems for Online Shopping. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1), February 2017. Number: 1.

[58] Shuo Zhang and Krisztian Balog. Evaluating Conversational Recommender Systems via User Simulation. *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1512–1520, August 2020. arXiv: 2006.08732.