Towards the Learning, Perception, and Effectiveness of Teachable Conversational Agents

by

Nalin Chhibber

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2019

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

The contents of this thesis has been adapted, revised, and extended from the following conference submission:

- Nalin Chhibber, Edith Law. **Using Conversational Agents To Support Learning By Teaching**. Presented at Conversational Agents Workshop, ACM CHI Conference on Human Factors in Computing Systems 2019.
- Edith Law, Parastoo Baghaei Ravari, Nalin Chhibber, Dana Kulic, Stephanie Lin, Kevin Daniel Pantasdo, Jessy Ceha, Sangho Suh, Nicole Belinda Dillen. **Curiosity Notebook: A Platform for Learning by Teaching Conversational Agents**, Submitted as Late Breaking Work to ACM CHI Conference on Human Factors in Computing Systems 2020.

The second submission mentioned above only contribute towards the system described in this thesis. Experiments are conducted in a different setting.

Abstract

The traditional process of building interactive machine learning systems can be viewed as a teacher-learner interaction scenario where the machine-learners are trained by one or more human-teachers. In this work, we explore if teachable AI agents can reliably learn from human-teachers through conversational interactions, how this teaching process affects a teacher's performance in the task, and their trust on the agent. We introduce a teachable agent named Kai, that learns to classify news articles while also guiding the teaching process through conversational interventions. In a three part study, where several crowdworkers individually teach Kai, we investigate whether this Learning by Teaching approach creates reliable machine learners, improves Turkers' performance and leads to trustable AI agents that crowdworkers would use. We present and discuss the results of the underlying classifier built from conversational interactions with other text classification algorithms. We also provide an evaluation of how crowdworkers perform a text classification before and after interacting with a teachable agent. Finally, we investigate the notion of trust that crowdworkers exhibit for their teachable agents in terms of delegating the work involving monetary compensation. Together, our results demonstrate the benefits of Learning by Teaching approach, in terms of the performance of the AI agent, the crowdworkers, and the dynamics of trust built from the teacher-learner interaction.

Acknowledgements

First, I would like to thank my family for their continued love and support. This would not have been possible without their motivation. I would also like to thank my supervisor Edith Law for giving me the opportunity to work on many exciting projects throughout my masters and for guiding my work along the way. Edith is an incredible person who is not only smart, but also very solicitous about her students.

In addition, I would like to thank Dan Vogel, and Charlie Clarke for agreeing to read my thesis and providing their valuable feedback. I have learned a lot from Dan and working with him has been a really joyful experience.

A special thanks to my friends Hemant, Rahul, Alex, Mike, Sangho, Jessy, Sasha, Greg and other friends from the HCI, AI and DGS lab for their unwavering support in academic and personal life.

Finally, many thanks to my sister Divya and members from the extended family in New Delhi and Brampton. Thank you all for being there when I needed it the most.

Dedication

I dedicate this to my parents Sh Ashok and Smt Reema Chhibber, who sacrificed everything for me, and believed in my abilities at times when even I didn't.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Recent progress in artificial intelligence has resulted in the development of intelligent agents that can direct their activities towards achieving a goal. Moreover, rapidly advancing infrastructure around conversational technologies has resulted in a wide range of applications around these agents, which include intelligent personal assistants (like Alexa, Cortana , Siri, and Google Assistant), guides in public places (like Edgar [50], Ada and Grace [126]), smart-home controllers [115], and virtual assistants in cars [86]. This growing ecosystem of applications supporting conversational capabilities has the potential to affect all aspects of our lives, including healthcare, education, work, and leisure. Consequently, agent-based interactions has attracted a lot of attention in HCI research [24, 89, 87, 85, 115]. The success of these agents will depend on their ability to efficiently learn from non-expert humans in a natural way. As these agents become more and more prevalent, it is important to explore certain interaction techniques that inform their development from HCI as well as AI perspective.

In this thesis, we investigate the domain of teachable agents, a special type of pedagogical agents that draws on the social metaphor of teaching a certain task to an autonomous or semi-autonomous entity. Conventional pedagogical agents are designed to play the role of teachers by giving instructions to the students. Here, we explore the domain of teachable conversational agents to study whether human-teachers can train a teachable AI agent through conversational interactions. Specifically, we investigate whether these teachable agents reliably learn from natural language conversations, how the teaching process affects human-teacher's performance later in the task, and whether human-teachers trust their own teachable agents. Understanding these questions can inform the design of teachable agents for various social interaction scenarios.

1.1 Thesis Objective

In this thesis, we present and discuss the results of a teachable agent that interactively queries crowdworkers in the process of building a text-classifier. We compare the results with traditional machine learning algorithms and examine how crowdworkers themselves perform a text classification task before and after interacting with this teachable agent. In addition, we investigate the notion of trust that crowdworkers would put on their teachable agents in terms of delegating the work involving monetary compensation. The high-level question we aim to investigate through our work is: can we design a teachable agent that crowdworkers can directly train, such that the agent can share their task workload. Specifically, we aim to answer the following research questions from our work:

RQ1: How well can crowdworkers train the teachable agent?
Traditionally, crowdsourcing is used for labelling machine learning datasets and crowdworkers are remunerated for their efforts through payments and bonuses. However, this does not often ensure meaningful engagement and leaves limited options to motivate the crowdworkers beside the monetary incentives. To address these concerns, we investigate the use of a teachable conversational agent that incorporates direct human teaching to interactively train a machine classifier. We hypothesize that compared to statistical machine learners:
H1: Teachable conversational agents representing an interactive machine learner, bootstrapped from statistical techniques will give comparable performance with meaningful engagement.

RQ2: What effects does teaching an agent have on the crowdworker's performance?
Protégé effect states that the best way to learn a concept is to teach it to someone else [25]. Traditionally, this effect has been mostly studied in peer-to-peer interaction scenarios occurring in classrooms or controlled laboratory settings. We intend to validate the effectiveness of this technique in the context of crowdsourcing platforms and hypothesize that:
H2: Crowdworkers who teach the agent will perform better than those who don't.

RQ3: Would crowdworkers actually adopt the teachable agent that they trained to share their work?
Establishing trust is a key requirement for the wide adoption and overall success of AI systems. Previous work has explored the use of algorithmic transparency (explainability), robustness, bias, privacy, reproducibility and accountability to build trust [22]. We posit

that personalizing the system and directly involving end-users in the training/teaching process can also strengthen their trust on AI systems. We hypothesize that:
H3: Crowdworkers will trust the agents taught by them by delegating the tasks.

1.2 Contribution

In summary, this thesis makes the following contributions:

- An interactive machine learning algorithm for text-classification that considers "statistical" as well as "user-defined" likelihood of words while predicting the posterior probability of a document belonging to a class.
- Performance comparison of the interactive machine learner built from conversational interactions with other algorithms for text-classification.
- Formalization of the learning-by-teaching paradigm and evaluation of its effectiveness in crowdsourcing tasks through teachable-agents.
- Investigation of trust on teachable-agents taught by online crowdworkers.

In addition, this thesis answers each of the research questions mentioned above, and discusses how teachable agents learn a task, what impact they have on the performance of humans teaching them, and how much they are trusted by the humans that teach them.

1.3 Organization

The remainder of the thesis is organized as follows:

- **chapter 2** provides background information on related work on interactive machine learning, conversational agents and agent-based interactions.
- **chapter 3** outlines the system description, provide details on the interactive machine learning algorithm and how dialog management and task environment.
- **chapter 4** describes the first crowdsourcing experiment investigating the interactive machine learning algorithm.
- **chapter 5** extends the experiment described in chapter 4 and investigates if the process of teaching a task to an agent improves crowdworkers performance in the task.
- **chapter 6** presents a final experiment that explores the dynamics of trust that workers put on the teachable agent for tasks involving monetary compensation.
- **chapter 7** concludes the thesis by summarizing the results and proposing directions for future work.

Chapter 2

Background and Related Work

In this chapter, we describe a brief overview of previous work on conversational agents within the HCI community. Then, we cover the research on agent-based interactions and how it relates to this thesis. Finally, we outline the domain of interactive machine learning that combines the efforts from human and the machine learning algorithm to address a problem. We conclude this section by describing machine teaching, that focuses on the human-centric aspect of interactive machine learners.

2.1 Research on Conversational Agents within HCI

Previous work on Conversational Agents is spread across various research themes. Recently, Clark et al. examined several topics in the HCI literature that are relevant to conversational agents, namely: system speech production, design insights, accessibility, and modality comparisons [31]. Work on system speech production has primarily studied the change in interaction behaviour of users based on specific manipulations to elements of (a) speech synthesis [99, 78, 37], (b) content of speech [66, 32], or (c) spatio-temporal aspects of the dialogues [67, 75]. Research related to design insights is focused on developing generally applicable guidelines for conversational interactions or iterations of developments for specific systems. This includes the designing of (a) iterative and bespoke systems tailored to accomplish specific tasks [57, 128, 127, 121], (b) dialogue modelling [40, 63] and (c) Automatic Speech Recognizers (ASR) [98, 109]. The research on accessibility has widely explored the use of conversational agents as assistive technologies that focus on users with specific requirements, or users interacting with specific type of systems [58, 107, 113]. Finally, the research on modality has explored both unimodal and multimodal systems

through speech and text, combined with more traditional input modalities such as keyboard [94], mouse [92], pen [104] and gestures [59]. Work related to modality comparisons is specifically relevant to our research. Le Bigot et al. found performance with an information retrieval system improved over time, regardless of using speech or text [76]. However, similar experiments comparing speech and written modalities observed lower efficiency when speech was used [77]. Hayashi and Ono studied the effects of modality on quality and quantity of interpretations in collaborative activities with embodied conversational agents [60]. Their results showed that while the use of text-based interfaces enhances the quantity of creative interpretations, voice-based interaction enhances the quality of results. Collectively, these results inform our decision to use textual modality over speech, to implement the conversational aspect of teachable agent as described in section 3.3.

2.1.1 Personalization of Conversational Agents

Personalization can be described as the process of making something suitable for the needs of a particular person [35]. In the context of information systems, it is defined as a process that changes the functionality, interface, information access and content, or distinctiveness of a system to increase its personal relevance to an individual or a category of individuals [48]. Fan and Poole proposed a framework to characterize personalization along three dimensions: (a) what is personalized (ie, content, user interface, functionality, and delivery channel); (b) for whom is it personalized (individuals or categories of individuals); and (c) how automated is the personalization (implicit or explicit) [48]. Personalization in Conversational Agents can be achieved implicitly by processing past interactions with users [123, 4] or explicitly by user-entered information at the set-up time [48] or using ongoing confirmation style input [26]. Past work has shown that Personalization of conversational systems can improve user comprehension [30], user satisfaction [100], task efficiency [16], and the likelihood of behaviour change [23].

In HCI research, personalization is often implemented as anthropomorphic software agents [7, 44]. Our work is inspired from the similar idea and attempts to emulate the teaching-learning scenario through conversational agents and interactive machine learning. Effectively, this inspired us to build Teachable Conversational Agents as a way to personalize the Human-AI interaction.

2.2 Agent Based Interactions

A significant amount of work in HCI and AI literature revolves around the notion of agents and agent-based interactions. However, the term "agent" has been often used in conflated ways and even regarded as the locus of considerable confusion [45]. First, it implies the existence of autonomous or semi-autonomous properties of AI systems like intelligence and responsiveness through *adaptive functionality*. Second, it suggests a particular model of what the program is, and how it relates to the user through the *agent-metaphor* [45]. While these two definitions often go together, the agent may not always comply with both. For instance, the agent reference used in many contexts such as "embodied conversational agents," "anthropomorphic interface agents", "virtual agents", or "pedagogical agents" etc, may not necessarily cover all aspects of adaptive functionality or the agent-metaphor. Further, the context may not require the Human-Agent interaction to be conversational in nature.

In this thesis, we specifically consider the agent contexts that encapsulate the teacher-learner interaction and adhere to adaptive functionalities through conversational interactions. One area that closely relates with our work is the domain of pedagogical agents. Pedagogical agents are lifelike characters presented on a computer screen that guide users through multimedia learning environments [29]. Their goal is to facilitate learner motivation and learning outcomes in such environments. However, most of the previous research in this area is focused on using agents for tutoring [53, 61, 5], question-answering [38, 49, 122], learning companions [80, 119, 14, 41], and dialogues to promote reflection and meta-cognitive skills [55, 72]. Moreover, these agents are mostly used as peers [111, 71], or tutors that play the role of a teacher or instructor [54, 95]. Our work is catered towards the scenario where these agents take the role of a less intelligent entity, allowing the students to teach [15, 17]. This approach is inspired from the Protégé effect, which demonstrates that learning for the sake of teaching others is more beneficial than learning for one's own self [25]. Previous work in cognitive science and education research supports the presence of the Protégé effect in reciprocal teaching [105], peer-assisted tutoring [33], small-group interaction [130] and self-explanation [27]. Studies focused on the cognitive benefits of teaching suggest that preparing to teach may produce more organized cognitive structures than learning the material for oneself [13]. Biswas et al. has shown that expecting to teach others helps in self-reflection, builds a sense of responsibility, and is useful for meaningful structuring of information [15]. This has been confirmed in later studies that demonstrate the effectiveness of the Protégé effect for cognitive [101], meta-cognitive [97] and motor learning skills [64]. Despite this, Protégé effect has not been fully explored with conversational agents or interactive machine learning systems. We explore this area within the

context of crowdsourcing systems as described in chapter 5.

2.2.1 Evaluation of Conversational Interactions

User interactions with conversational agents have been measured through both objective as well as subjective metrics across various dimensions like task performance, user attitudes, perceived usability, system usage, and cognitive load. Task performance has been measured using total number of conversational turns [77], percentages of tasks completed correctly [104], and task completion time [106]. Attitude of users towards conversational agents has been studied by measuring likeability and human likeness [32]. Perceived usability has been mostly examined through scale based questionnaires on perceived ease of use and learnability [46]. System usage has been quantified to study what people use conversational interfaces for [36] and how they used them [114]. Finally, studies on measuring cognitive load deals with identifying physical, mental and temporal demands of users while interacting with conversational systems. Informed by these methods to evaluate the interactions with conversational systems, we employ a set of subjective as well as objective measures in order to quantify the effectiveness of teachable conversational agents.

2.3 Interactive Machine Learning

Traditional machine learning aims to solve problems without any human intervention. These algorithms are used to create predictive models based on a given dataset. However, prediction accuracy of these models can only be as good as the quality of the training data used. Otherwise, the model can make incorrect predictions, or take more time to learn. In contrast, interactive machine learning attempts to overcome these problems by involving users directly in the process of optimizing the machine learning models. It allows rapid, focused and incremental updates to the model, thus enabling users to interactively examine the impact of their actions and adapt subsequent inputs to obtain desired behaviours. In essence, interactive machine learning facilitates the democratization of applied machine learning by allowing humans to interact with machine-learning-based systems to address a problem.

One of the earliest work in this area is from Ankerst et al. who worked on an interactive visualization of classification tree [9]. They created an interface that provide sliders to adjust the number of features or threshold values for each node in the decision tree, and interactively display the classification error. Ware et al. demonstrated that humans can

produce better classifiers than traditional automatic techniques when assisted by a tool that provides visualizations about the operation of specific machine learning algorithms [129]. Within the HCI community, work on interactive machine learning was first explored by Fails and Olsen [47]. They studied the difference between classical and interactive machine learning and showed an interactive feature selection tool for image recognition. Fiebrink et al. created a machine learning system that enable people to interactively create novel gesture-based instruments [51]. Their experiments found that as users trained their respective instruments, they also got better and even adjusted the goals to match observed capabilities of the machine learner through the interactive nature of the system. These examples illustrate how rapid, focused and incremental interaction cycles can facilitate end-user involvement in the machine-learning process. Porter et al. formally break down the interactive machine-learning process into three dimensions: task decomposition, training vocabulary, and training dialogue [108]. These dimensions define the level of coordination, type of input, and level/frequency of interaction between the end-users and machine learners. Later, some researchers examined the role of humans in interactive machine learning, and highlighted various areas where humans have interactively helped machine learning systems to solve a problem [6]. Most of the work in this area suggests a diversity in terminologies used across different disciplines. This informs the need to develop a common language to accelerate the research on interactive machine-learning systems from both machine-centric and human-centric perspectives.

2.3.1 Active Machine Learning versus Machine Teaching

Active learning is a special case of interactive machine learning that focuses on improving machine learner's performance by actively querying a human oracle and obtain labels [116]. These labels are obtained through various querying strategies that can select instances which (a) the machine learner is least certain about (Uncertainty Sampling) [81], (b) creates maximal disagreement between multiple models being trained (Query By Committee) [118], (c) have greatest influence on the model (Expected Model Change) [117], or (d) reduce the expected generalization error (Expected Error Reduction) [112]. Some of the first active learning scenarios were investigated by researchers during late 80's and early 90's [8, 11, 34]. Since then, researchers have explored the use of active learning with support vector machines [125], Bayesian networks [124], named entity recognition [102], and natural language processing [103]. However, several studies reveal that active learning can cause problems when applied to interactive settings [20, 18, 56]. One of the primary challenge with active learning in such situations is: humans are not always willing to be simple oracles by answering a stream of questions through traditional user-interfaces. To address

these concerns, many researchers have recently started focusing on the human-centric part of these interactive systems.

Machine teaching is a discipline that focuses on the efficacy of teachers and their interaction with data [120]. While machine learning focuses on creating new algorithms and improving the accuracy of "learners", machine teaching measures performance relative to human costs, such as productivity, interpretability, robustness, and scaling with the complexity of the problem or the number of contributors [120]. The primary inspiration behind this research is the idea that a helpful teacher can significantly improve the learning rate of a machine learning algorithm, as shown in the field of Algorithmic Teaching [12, 90, 52]. Simard et al. formalize the role of teachers as the humans who transfer knowledge to machine learners so that they can generate useful models to approximate a concept [120]. However, human teaching is mostly optimized for human learning, and therefore is not naturally optimal for arbitrary machine learners. Cakmak et al. investigated ways to elicit good teaching from humans for interactive machine learners [21]. They propose the use of teaching guidance to let human teachers adapt for the needs of specific machine learners. Teaching guidance is a set of instructions given to human teachers, that influences their choice of examples towards most informative ones for a particular learner [21]. This can be in the form of either algorithm or a heuristic based guidance. While algorithms can have guaranteed optimality bounds, they are often not as amenable to be used as teaching guidance. On the other hand, although heuristic-based guidance may not guarantee optimality, they are often easier to understand and use for everyday people [21]. Heuristic-based teaching guidance has also been used for inverse reinforcement learning agents in the sequential decision making tasks [19]. Our work is partly inspired from the machine teaching philosophy, and demonstrates the use of conversational interactions to elicit information from human teachers in chapter 4.

2.3.2 Summary

In summary, this section covered the previous work related to agent-based interactions and interactive machine learning in order to justify the use of these agents as an interface between the humans and a traditional machine learning algorithm. Specifically, our work leverages the conversational interface to elicit teaching from the humans through the principles of machine teaching and use the captured information to improve the performance of an underlying machine learning algorithm. Rest of this thesis covers the system description, experimental results and discussions on the design implications for interactions related to teachable conversational agents.

Chapter 3

Dataset and System Description

In this chapter, we describe the dataset and system used to facilitate the interaction between the human-teacher and a teachable machine learner. Then, we discuss the classification algorithm that combines input from human-teachers and statistical techniques to classify text documents. Lastly, we describe the conversational interface and task environment used to conduct the experiments.

3.1 AG News Classification Dataset

We used AG News Classification Dataset [1], which is consisted of more than 1 million news articles gathered from more than 2000 news sources by an academic news search engine called ComeToMyHead [2] since July, 2004. AG news dataset was used because it has been used as a benchmark by previous work on text classification. The dataset was made available to the academic community to do research in data mining (clustering, classification, etc), information retrieval (ranking, search, etc), xml, data compression, data streaming, and any other non-commercial activities.

For news topic classification, we used the dataset constructed by Zhang et al. [131], with 4 largest classes representing the topics World, Sports, Business and SciTech. Each class contains 30,000 training samples and 1,900 testing samples. The total number of training samples in the dataset is 120,000 and number of test samples is 7,600.

Data Preprocessing

We followed a series of steps to pre-process the data obtained from AG News classification dataset and use it to train the classifier. These steps are described as follows:

- **Tokenization**: Tokenization is the process through which text is stripped out to a simpler sets of essential units called tokens. These tokens are either created at word level or sentence level. Words tokens are useful for finding patterns in the text and considered as a base step for stemming and lemmatization. We used the *word_tokenize*() function from NLTK to split the raw sentences from dataset into separate word tokens. This process was followed by a text normalization step where we converted individual tokens into lowercase to maintain the consistency during training and prediction.

- **Stop Words Removal**: Stop words are the commonly used words (such as 'the', 'an', 'of', etc) that do not contribute much to the context and semantics of the text when it comes to classification techniques. These words hold almost no importance for the purposes of information retrieval and natural language processing and often add noise to the text being analyzed. Therefore, it is important to remove such words from extracted tokens before further analysis. NLTK python package comes with many stopwords corpus (*nltk_data/corpora/stopwords/*) that contain word lists for different languages. We used the list of stopwords from NLTK English corpus to filter out the words that did not contain vital information for text classification.

- **Lemmatization**: Different languages may have different degrees of inflections depending on how the words are modified to express different grammatical categories such as tense, case, voice, aspect, person, number, gender, and mood. Lemmatization is a text normalization technique that is used to handle such language inflections by converting words to their base form. It is similar to other text normalization techniques like Stemming; however, Lemmatization considers the context while performing the base form conversion, while stemming performs suffix-stripping and does not ensure that the stem (root) is a valid word in the original language. We used WordNetLemmatizer with with POS tags to obtain the canonical form (lemmas) of the tokens.

3.2 Classification Algorithm

Classification is a predictive modelling problem that involves assigning a label to a given input data sample. From the Bayesian perspective, classification algorithms are categorized into discriminative or generative models based on how they estimate the conditional probability of the output class. Discriminative classifiers are known to predict the posterior probability $P(y|x)$ directly as a mapping from the input x to the class label y. Generative classifiers on the other hand, learns the joint probability, $P(x, y)$ of inputs and the output labels, and make their predictions by using the Bayes rule to calculate $P(y|x)$ and then picks the most likely label y. In this work, we use generative classifiers with Bayesian inference to learn text classifications with additional inputs from conversational interactions. The actual classification on new instances is performed using Bayes theorem by selecting the class with the largest posterior probability.

3.2.1 Naive Bayes Classifier

Naive Bayes is one of the simplest Bayesian classifiers; it applies the Bayes' theorem while making strong assumptions that the features are conditionally independent.

Bayes Theorem

In probability theory and statistics, Bayes theorem describes the probability of an event based on prior knowledge of conditions that might be related to the event.
Mathematically, Bayes' theorem is stated as the following equation:

$$P(A|B) = \frac{P(A)P(B|A)}{P(B)} \tag{3.1}$$

Where, A and B are events and P(B)$\neq$ 0.

- $P(A|B)$ is the conditional probability of event A occurring given that event B has already occurred.
- $P(B|A)$ is the conditional probability of event B occurring given that A has already occurred.
- $P(A)$ and $P(B)$ are the probabilities of observing events A and B respectively.

In simple terms, (3.1) can be expressed as:

$$Posterior = \frac{Prior \times Likelihood}{Evidence} \tag{3.2}$$

Here, **prior** and **evidence** refer to the probabilities of observing A and B independently from each other, whereas the **posterior** and **likelihood** are the conditional probabilities of observing A given B, and vice versa.

Probabilistic Model

Naive Bayes classifier is a probabilistic machine learning model that uses Bayes theorem to predict posterior probability of an event. It assumes that the presence of a particular word in a class is independent to the presence of any other word in that class. Furthermore, Naive Bayes predicts a probability distribution over a set of classes instead of merely outputting the most likely class. Accordingly, under the Naive Bayes model, Bayes theorem can be rewritten as:

$$P(C_k|\mathbf{w}) = \frac{P(C_k)P(\mathbf{w}|C_k)}{P(\mathbf{w})} \tag{3.3}$$

Here the variable C_k represents a document class from (World, Sports, Business, or SciTech). Variable **w** represent the feature vector containing words from the respective document and mathematically defined as:

$$\mathbf{w} = (w_1, w_2, w_3 ... w_n) \tag{3.4}$$

Here, $w_1, w_2, w_3 ... w_n$ represent the individual words coming from document class c. Overall, equation (3.3) describes $P(C_k|\mathbf{w})$ as the **posterior** probability of a document belonging to a class given its constituent words, in terms of $P(\mathbf{w}|C_k)$: the **likelihood** of words coming from a known class, $P(C_k)$: **prior** probability of the class distributions and the **evidence** term represented by $P(\mathbf{w})$. Due to the assumption of feature independence, **likelihood** can be calculated as the product of the individual probabilities of seeing each word in the set of documents. Formally,

$$\begin{aligned} P(C_k|w_1, w_2 ... w_n) &\propto P(C_k) \prod_{i=1}^{n} P(w_i|C_k) \\ &= P(C_k)\ P(w_1|C_k)\ P(w_2|C_k) \dots \end{aligned} \tag{3.5}$$

The proportionality symbol in (3.5) is introduced due to the independence assumption and removal of the **evidence** term as it is a constant factor depending only on words.

The **prior** probability of document classes is calculated by either assuming equi-probable classes ($prior = 1/\text{number of classes}$), or by calculating an estimate for the class probability from the training set ($prior_c = (\text{number of samples in class c})/(\text{total number of samples})$). Finally, the **likelihood** is estimated by assuming a distribution or by assuming a non-parametric model for the words in the training set. The assumptions made on the distribution of words depend on the event model of the Naive Bayes classifier. For continuous features, values associated with each class are assumed to fit a Gaussian distribution, resulting in what is known as a Gaussian Naive Bayes classifier. For discrete features like words occurring in a document, Multinomial and Bernoulli distributions are used, resulting in two distinct variants of Naive Bayes.

Bernoulli Naive Bayes

In the multivariate Bernoulli event model, features are represented as a binary variables describing the occurrence of a word in all documents from a given class. If w_i is the i^{th} term from the vocabulary, then its conditional probability given the class is expressed as:

$$P(w_i|C_k) = P(w_i|C_k)^{x_i}(1 - P(w_i|C_k))^{1-x_i} \tag{3.6}$$

where x_i is a boolean expressing the presence or absence of the i'th term from the vocabulary.

Bernoulli Naive Bayes is popular for classifying short texts and have the benefit of explicitly handling the absence of terms. However, it does not capture the information about the number of times a word occurs in a document.

Multinomial Naive Bayes

In the multinomial event model, features are represented as the frequency with which certain events are generated in a multinomial distribution. This property makes it particularly useful for document classification as it can represent the frequency of occurrence of a word in a document. This model for text classification was introduced by McCallum et al. [91] as an improvement over Multivariate Bernoulli model for long text documents. Here, conditional probability of each word given the class is expressed as follows:

$$P(w_i|C_k) = \frac{count(w_{ik})}{count(w_k)} \tag{3.7}$$

Where $count(w_{ik})$ is the number of times word i from the test document appears in class k across all the samples in training data, and $count(w_k)$ is the total number of words in class k in training data. A Multinomial Naive Bayes classifier can be treated as a linear classifier when expressed in log space [110].

One of the common challenges encountered in Multinomial Naive Bayes is the handling of 0 probabilities from features that do not exist in the vocabulary. If not handled, these features can potentially wipe out all the information from other feature probabilities when they are multiplied. To avoid this, Lidstone smoothing is applied as a regularization technique for all classes $(C_1, C_2..C_k)$ as described below:

$$P(w_i|C_k) = \frac{count(w_{ik}) + \alpha}{count(w_k) + \alpha \ |k|} \tag{3.8}$$

Modified Naive Bayes Classifiers

Naive Bayes classifiers are known to perform well for many classification tasks even when the conditional independence assumption on which they are based is violated. Domingos et al. have discussed the feature independence assumption and explained why Naive Bayes performs well for classification even with such a gross over-simplification [43]. On the flip side, many of the studies reporting superior performance of Naive Bayes classifiers are focused on smaller datasets with balanced classes. It has been shown that the classification accuracy of Naive Bayes does not scale well with large data sets [73]. Rennie et al. discussed ways to improve the accuracy of multinomial models through tf–idf weights instead of raw term frequencies and document length normalization [110]. Another popular technique to boost the classification accuracy is to relax the conditional independence assumption through locally weighted learning [10]. Past work has thoroughly discussed local likelihood methods like locally weighted linear logistic regression [84], locally weighted density estimation [84, 28] and locally weighted decision trees (C4.4) [70]. Frank et al. proposed a similar algorithm—a locally weighted Naive Bayes that weighs the k nearest neighbours of the test instance in terms of their distance to that instance. This helps to weaken the e ffects of attribute dependencies that may strongly exist in the whole training data but much weaker within the neighbourhood of the test instance. Thus, most of the work on improving the classification performance of Naive Bayes is focused on addressing the independence condition either explicitly by directly estimating dependencies, or implicitly by increasing the number of parameters to estimate.

3.2.2 Proposed Approach

As described above, there are many variants of Naive Bayes classifier that aim to improve its classification performance by making certain assumptions about the distribution of words, or relaxing the independence assumption. In this thesis, we adopt the idea of estimating dependencies for features in the test document and relaxing the feature independence assumption through a human-in-the-loop system. The idea is to infer the class of a test document given its words, through the conditional probabilities of having those words present in training data and/or similar words captured from conversational interactions with a human-teacher on each of the available classes (C_k). Given the set of words from a test document, the conditional probability for those words in training data under respective classes is represented as $P(w_i|C_k)$ and the conditional probability of conversational keywords that are similar to the words in the corpus is represented as $P(s_i|C_k)$.

$$P(s_i|C_k) = \frac{\text{\# conversational keywords similar to } word_i \text{ in test document}}{\text{Total \# conversational keywords captured from the interaction for C}_k} \quad (3.9)$$

To determine whether a conversational keyword is similar to a word in the test document, we calculated the average cosine similarity between the words appearing in the test document and the words discussed during conversational interactions for each document class. Cosine similarity measures the cosine of the angle between two non-zero vectors in the same vector space, representing their closeness. Since cosine similarities can only be calculated for vectors, we transformed the words into word-embeddings using Word2Vec model. Word2Vec is a shallow neural-network that is trained to reconstruct the linguistic contexts of words in vector space [93]. The original pre-trained model is trained on 3 million 300-dimension word vectors in English language. We used a lighter model with same dimensionality of vectors as the original model but trained on 300,000 words from Google News dataset, crossed-referenced with English dictionaries. Similarity coefficients are calculated using the Gensim package that provides a Python implementation of Word2Vec with an in-built utility for finding similarity between two words given as input. Note that these similarity coefficients can only be calculated if both words are present in the Word2Vec model vocabulary: words from the conversation and the words from the test document. We applied Laplace smoothing to both conditional probabilities for reasons described above by setting $\alpha = 1$ in equation (3.8). These similarity coefficients are then multiplied with the statistical likelihood of corresponding words appearing in the training corpus in order to calculate the overall posterior probability of the text belonging to a specific document class. Similarity coefficients have a range between -1 and 1, with negative values indicating that the words are not similar to each other, and positive values

indicating greater similarity between the words. Conversational keywords where the similarity coefficient is below a threshold (e.g. 0.2) are not counted in (3.9). The similarity coefficients between the words in test document and words obtained from conversations can be determined through any other distance metric technique like Minkowski distance, or Jaccard scores. We refer these similarity coefficients as user-defined likelihood and use them to to modify the posterior probabilities of a test document belonging to a class based on the following two situations.

Case 1: Without supervised pre-training

In this case, the posterior probability of the document belonging to a class is only inferred from the conditional probability of the conversational keywords captured during the discussion. Thus, equation (3.5) can be expressed as:

$$\begin{aligned} P(C_k|w_1, w_2...w_n, s_1, s_2...s_n) &\propto P(C_k) \prod_{i=1}^{n} P(s_i|C_k) \\ &= P(C_k)\ P(s_1|C_k)\ P(s_2|C_k) \dots \end{aligned} \tag{3.10}$$

Case 2: With supervised pre-training

In this case, the conditional probability of the conversational keywords captured during the discussion is combined with conditional probability of the words in the original corpus. Thus, equation (3.5) can be expressed as:

$$\begin{aligned} P(C_k|w_1, w_2...w_n, s_1, s_2...s_n) &\propto P(C_k) \prod_{i=1}^{n} P(w_i|C_k) P(s_i|C_k) \\ &= P(C_k)\ P(w_1|C_k)\ P(s_1|C_k)\ P(w_1|C_k)\ P(s_2|C_k) \dots \end{aligned} \tag{3.11}$$

Note that the conditional probability of a word appearing in the training corpus, $P(w_i|C_k)$, and the conditional probability of similar words being discussed during the conversational interaction, $P(s_i|C_k)$ are considered as two independent events and hence their combined probabilities can be expressed as the product of individual probabilities.

Decision Rule for Classification

To yield the final classification, the algorithm outputs the class with the highest posterior probability:

$$y = \text{argmax } P(C_k) \prod_{i=1}^{n} P(w_i|C_k)\ P(s_i|C_k) \tag{3.12}$$

3.3 Conversational Interface

3.3.1 Why a Conversational Interface?

Previous research on agent-based interactions has found that the mere presence of a lifelike character in an interactive learning environment induces strong positive effect on learner's perception of the learning experience (persona effect) [79, 96, 42]. However, a later study found the text-based interface delivering third-person references to a subject to be more enjoyable leading to better learning outcomes in comparison to a conversational agent delivering first-person references (Piagetbot) [62]. Nevertheless, most of the work advocating non-conversational interactions has focused on agents that are mostly used as peers [111, 71], or tutors playing the role of a teacher or instructor [54, 95]. Since our work focuses on presenting the agent as a less intelligent entity, an important benefit of using the conversational interface is that clarification can be obtained through the interaction process. Further, using a conversational interface to obtain the user-defined likelihood of features is useful because it can address some important concerns that directly relate to the performance of Naive Bayes classifiers:

- **Independence condition**: Performance of the Naive Bayes classifier degrades when the attribute-independence condition does not hold. This limitation can be addressed by relaxing the independence assumption through additional features that are discussed during conversational interactions on a given topic.
- **Limited training data**: Although Naive Bayes classifiers are known to perform well with limited training data [65], their performance still depend on the size and richness of feature vocabulary. By incorporating additional features from conversational interactions, we can account for words in the test document that do not appear in the original training data.

- **Imbalanced classes**: Performance of a Naive Bayes classifier can be skewed towards a class with significantly higher number of training samples. Although past work has addressed this issue by applying tf-idf weights to features [110], it still does not account for limited feature vocabulary from classes with lesser training samples. A teachable conversational agent can conduct more discussions on topics that have lesser training samples and capture new features for those classes.

Interaction Modality

Previous work on the modality comparisons in conversational or pedagogical agents is full of mixed results. More recent studies on embodied conversational agent show that while the use of text-based interfaces enhances the quantity of creative interpretations, voice-based interaction enhances the quality of results [60]. Moreover, using a textual interface over voice helped us avoid errors originating from incorrect speech recognition and challenges in speech-to-text conversions.

3.3.2 Dialog System

Conversations in the dialog system were designed using a dialogue tree, a branching data structure. Dialogue trees are popular in game design and similar to story trees where each node represents a place where a conversation may branch, based on the users' decision about what they want to say [3]. Unlike a story tree, links in a dialogue tree can go backward or forward because of the nature of conversational interaction (eg. repeating a sentence). Besides managing the conversation, we kept a separate strategy to manage states and overall interaction of the agent with human teachers.

State Management

The dialog states are managed through a rule-based approach built using a hierarchical state machine. Hierarchical state machines are the finite state machines whose constituent states themselves can be other state machines. The top-most level of our dialog system hierarchy represents the mode of agent's primary interaction with the human-teacher. This mode could either be the learning mode, or the validation mode. In learning mode, the teachable agent is focused on learning new features through conversations related to a given topic. In validation mode, agent attempts to predict the category of unseen news snippets and asks for more samples to predict from the human-teachers. The agent can switch

between either modes based on the interaction sequence determined by human-teacher. This interaction sequence to switch modes can be triggered by either pressing a button, giving a verbal command or touching a sensor based on the interaction modality of the teachable agent. Each of these modes further contain multiple contexts that defines the second level of our dialog system hierarchy. The context refers to the states that describe the relevance of features for a given topic. The agent can switch between different contexts in order to capture new features that are relevant or irrelevant to the topic under discussion. We used these contexts to represent different teaching heuristics aimed to facilitate the teaching process for humans. These teaching heuristics are explained in detail in the following subsection. Finally, all contexts contain multiple intents that decide the sequence of conversation. These intents can either be recognized through rule-based heuristics or deep learning methods. We used the rule-based approach to identify different intents during the conversational interactions. In addition, we also developed agent strategies loosely consistent with Speech Act theory that direct the user to ask about content within Kai's dialog system repertoire. In certain cases in which no input was recognized, Kai would default to one of several fallback options like: asking users to paraphrase, repeat or simply ignore and move to next .

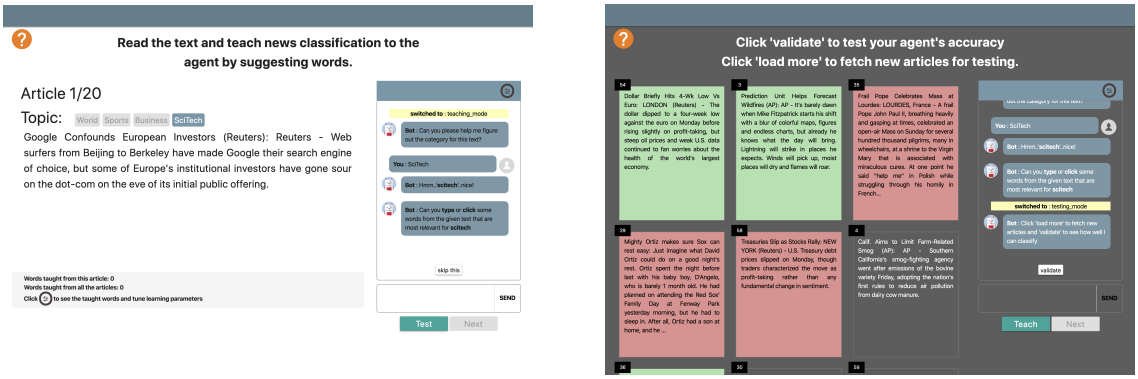
3.3.3 Teaching Guidance

Heuristic	**Instruction**	**Conversational Guidance**
1. Internal relevant words	1. Select few words from the text that are most relevant to the *category*	I wonder which words are most relevant while categorizing this text to the *category*?
2. Internal irrelevant words	2. Select few words from the text that are least relevant to the *category*	Which words are least relevant while categorizing this text to the *category*?
3. External relevant words	3. Enter few words 'outside' the text that will most likely describe the *category*	Can you tell me few more words that should describe the *category* but are not in the text?

Table 3.1: Heuristics with corresponding instructions and teaching guidance given to crowdworkers during the task

Past work on algorithmic teaching has shown that human teachers can significantly improve the learning rate of a machine learning algorithm [12, 90, 52]. However, humans often do not spontaneously generate optimal teaching sequences. Moreover, human

teaching is mostly optimized for human learning and therefore not naturally optimal for arbitrary machine learners. Cakmak et al. examined several ways to elicit good teaching from humans for machine learners [21]. They proposed the use of teaching guidance based on computational solutions to the teaching problem at hand which can either be in the form of an algorithm or a heuristic. As described above, while algorithms can have guaranteed optimality bounds, they are often not as amenable to be used as teaching guidance. On the other hand, although heuristic-based guidance may not guarantee optimality, they are often easier to understand and use for everyday people. Macgregor et al. proposed two teaching heuristics for optimizing the classification algorithms [88]. Similar to their approach, we identify three teaching heuristics that may help our interactive Naive Bayes classifier and also amenable for humans. Features identified through these heuristics were meant to supplement the classifier by proposing new features, amplifying relevant ones, or discounting the irrelevant ones for respective categories. Table 3.1 summarizes the heuristics with corresponding instructions and teaching-guidance. This teaching guidance was used to form the second level of our dialog system hierarchy described above.



(a) Teaching Interface (b) Testing Interface

Figure 3.1: Task Environment: Curiosity Notebook

3.4 Task Environment

We used the task environment provided by Curiosity Notebook: a learning-by-teaching platform with conversational agents. Curiosity Notebook provides a web-based learning environment that supports the interaction between student-teachers and agent-learners. The purpose of this system is to focus on the nature of the interaction between a tutor

and tutee. Specifically, we aim to promote the scenario that enables students to take the role of an instructor who teaches a classification task to a virtual conversational agent. We chose classification tasks because they are well structured which means that the teaching conversation can be designed to be highly structured as well. Classification tasks are also amenable to machine learning, allowing computational models of learning to be eventually implemented in the agent as described above. Although our previous studies on Curiosity Notebook were focused on classifying objects that involves mainly identifying and remembering features that distinguish each category, our work in this thesis is focused on facilitating text-classification. Additionally, unlike previous experiments where Curiosity Notebook was used along with a physical humanoid robot in controlled lab and school studies, we decided to run our experiments online on Amazon Mechanical Turk for multiple reasons. First, it allowed to to quickly prototype the system and test it with small sample groups. This allows frequent iterations necessary to build a robust experimental prototype. Second, it broadens the sample population that includes participants from a diverse set of age group, gender, and geographic regions. Previous research have shown that crowdsourced experiments and experiments yield equally valid results [74]. In essence, Curiosity Notebook was extended to facilitate online teaching in crowdsourcing context.

In the task interface, participants could switch between Teaching and Testing modes to either teach their agent, or validate their performance on a set of unseen articles. These two modes are described in Figure 3.1. Within the teaching mode, while reading the article, participants could highlight sentences and use them to direct conversations in natural language dialogues through a textual interface (Figure 3.1(a)). The agent then asks questions and prompts the human-teacher to elicit their queries and reveals what it does not understand about the topic, or what else it wants to know. These queries are answered by the human-teacher interacting with the agent, allowing them to reflect upon their own knowledge and ultimately gain a better understanding about the topic. In the testing mode, participants could load new articles in the interface and ask the teachable agent to classify them in real-time based on what they have learned from the conversational interaction. After the agent's prediction, correctly classified articled were coloured green, whereas incorrectly classified articled were coloured red. During the entire interaction, participants were encouraged to switch between teaching and testing modes in order to check their teaching performance and how the agent adopts to it with new articles and additional words.

Chapter 4

Experiment 1: Formative Evaluation

The purpose of this experiment is to test the learning algorithm described in previous section with a group of crowdworkers. The aim is to evaluate the performance of Naive Bayes classifier built from the keywords gathered from the conversational interaction with and without using the supervised pre-training. We also compare the performance of our classifier with the baseline text classification algorithms that were modified. Results from this study not only helped us gather necessary conversational data to evaluate our system, but also provided useful insights into how such interactions should be modelled into conversational setups in order to maximize agent's learning performance. Additionally, we also capture user-feedback on the Computer System Usability Questionnaire to evaluate their opinion on overall satisfaction with a system having teachable agent. Thus, this study serves as an important part to evaluate the learnability of Teachable AI systems that learn a task through conversations. Figure 5.1 describes the general experimental procedure.

4.1 Design

We conducted a formative study to investigate whether our modified variant of Multinomial Naive Bayes classifier can iteratively learn from additional keywords captured from conversational interactions. For this, we designed an experiment within Curiosity Notebook where a human teacher reads a series of news articles and help the conversational agent embedded as a chatbot to learn the news classification. The chatbot asks questions related to the articles and capture the user-utterances made in response. Each human teacher teaches their own version of the agent that allows us to evaluate agent's learning

progress after each article as an independent epoch. Dependent measures from the study are as follows:

- **Words taught**, the total number and proportion of words taught to the agent.
- **Change in agent's performance**, the classification performance of the agent as it learns from human-teachers without supervised pre-training.
- **Overall agent's performance**, the overall classification performance of the agent after the conversational interaction combined with supervised pre-training.

Besides these dependent measures, we also measured participants' responses to the post-study questionnaire, including their opinion on the overall interaction and feedback on the systems involving teachable agents.

4.2 Participants

We recruited sixty crowdworkers from Amazon Mechanical Turk (10 females, 50 males), 23 to 53 years old (M= 30.9, SD= 5.29). The study was conducted by posting Human-Intelligence-Tasks (HITs) with the title: "Teach How to Classify News Articles to a Chatbot". Participant pool represented a variety of professions including managers (13), IT technicians (10), engineers (9), clerks (5) , analysts (4) and designers (3). Three of the participants were teachers, two were homemakers and remaining were self employed. 87% of the participants were native English speakers, but all reported some prior experience with conversational agents on a 7-point scale (M=5.76, SD=1.15). 53.4 % of the participants reported prior experience in teaching a classification to someone else, the other half had no prior experience on teaching (46.6%). Regarding the prior knowledge on the 4 given news categories, participants rated most for World (M=5.85, SD=1.20), followed by SciTech (M=5.63, SD=1.27), Business (M=5.55, SD=1.47) and Sports (M=5.07, SD=1.78).

Participants received $0.5 USD for the pre-study questionnaire on demographics, $2 USD for the primary teaching task and $0.5 USD for the post-study questionnaire. The teaching task took approximately 30 minutes, pre- and post-study questionnaires took 2-5 minutes each to complete. The experiment was conducted within Curiosity Notebook running on Django framework in the backend and Javascript in the frontend. Only participants using Chrome and Firefox were allowed to participate in order to reduce the possibility of browser incompatibility.

4.3 Procedure

The experiment was hosted as an independent Web Application running on Python Django framework. The workers were first meant to accept the HIT from Amazon Mechanical Turk and click a link that would open the application interface in a new browser window. Upon completion, the application generated a random alphanumeric token which was meant to be entered in the Mechanical Turk Interface while submitting the HIT. The general procedure adopted for all studies is described in Figure 4.1

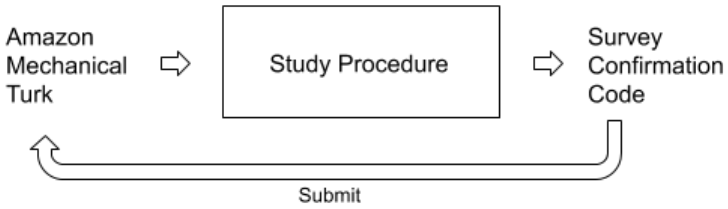


Figure 4.1: General experimental procedure for MTurk studies

In the experiment, workers first read the information and consent letter explaining the details about the study. Then, after providing the consent for participation, they were given a short tutorial on the interface explaining different UI elements and their usage. After this, they were shown a news article and a chat interface to teach the classification to the virtual teachable agent. The agent used to ask the category for the given article and ways to classify it into one of the known categories. During the teaching process, workers were free to switch between the "Teach" and "Test" mode by clicking respective buttons below the chatbox. In the test mode, the agent would predict the category of the articles based on words that were taught during the "Teach" mode. In total, there were 20 articles to teach that were equally distributed across all four news categories. Workers were supposed to teach at least one word from each article in order to proceed further in the Task. The study procedure and teaching interface are shown in Figure 4.2 and 4.3 respectively.

After completing the teaching task, workers were asked to fill the IBM Computer System Usability Questionnaire (CSUQ) [83] to report their opinion on the overall interaction experience and satisfaction with the system having a teachable agent.

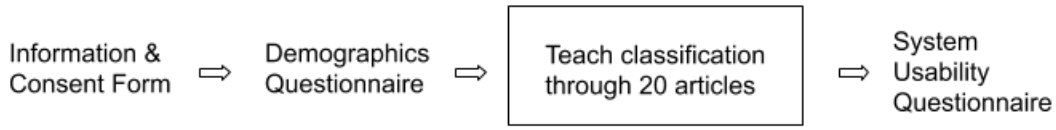
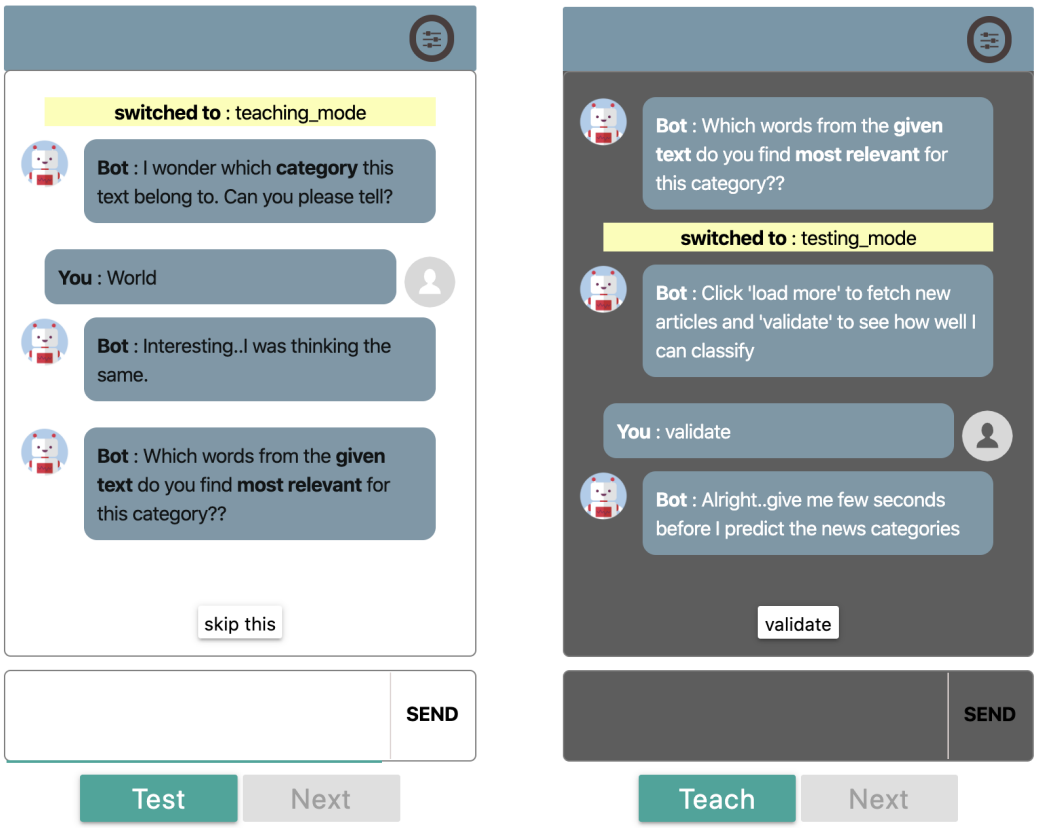


Figure 4.2: Study procedure for experiment 1



(a) Teaching (b) Testing

Figure 4.3: Interaction with the agent during (a) teaching, and (b) testing mode

4.4 Analysis Methods

This section describes the metrics used to evaluate engagement of crowdworkers during the experiment and measure the classification performance of agent.

4.4.1 Indicators of crowd performance

We analyzed the data gathered from each crowdworker in the pre-study questionnaire as well as the final performance of the agent that was taught by them. This was done to examine whether there is any relation between these pre-experiment attributes and worker's ability to teach a classification task to the agent. If such a relationship exists and is significant, it can be used as a mechanism to filter out poor teachers for subsequent studies.

4.4.2 Teaching Efforts

Teaching efforts were measured by recording the entire transcript of the conversation between the crowdworkers and the teachable agent. From this, we inferred the total time spent on teaching, number of dialogues exchanged during the interaction, number of words that were taught to the agent, and number of times the agent was tested by the crowdworkers. This was done to understand what factors from worker's ability to teach a classification task are most relevant to the final classification performance of the agent.

4.4.3 Agent's Performance

Agent's performance was evaluated by measuring its classification performance (F-1 scores) on the test set. For the experiment, we used the interactive variant of Multinomial Naive Bayes as the underlying classification algorithm. The interactive Naive Bayes was used without supervised pre-training (as described in equation (4.3)) in order to minimize the confounds resulting from initial performance being too high. It was necessary because an interactive classifier with supervised pre-training would already be good in news classification before the conversational interaction. This could have primed the participants to not teach as many words as they may teach otherwise, assuming that the agent is already classifying news-snippet accurately. Other variants of the interactive Naive Bayes were later analyzed offline to identify the most useful algorithm that can learn from human-teachers

during the experiment. The metrics used to quantify the classification performance of the teachable agent are described below.

Evaluation Metrics

In order to evaluate the performance of the agent representing a text-classifier, we computed precision (P), recall (R) and F1-score (F1) by comparing the predicted output with ground-truth answers for the news articles in the test-set. These measures are defined as follows:

- Precision: It attempts to predict the proportion of positive identifications that were actually correct (TP). Thus, a model that does not produce any false positives (FP) has a precision of 1.0.

$$P = \frac{TP}{TP + FP} \tag{4.1}$$

- Recall: It attempts to predict the proportion of actual positives that were identified correctly (TP). Thus, a model that does not produce any false negatives (FN) has a precision of 1.0. Recall is also called referred as Sensitivity of a model.

$$R = \frac{TP}{TP + FN} \tag{4.2}$$

- F1-Score: Sometimes, it is desirable to consider both precision and recall in order to fully evaluate the effectiveness of a model. However, both these metrics are often inversely proportional to each other and improving precision reduces the recall and vice versa. In such cases, the most effective metric to evaluate the model performance is to find F1-score by taking the harmonic mean of Precision and Recall. Thus, F1-score conveys a balance between the precision (P) and the recall (R).

$$F1 = \frac{2PR}{P + R} \tag{4.3}$$

We used F1 scores to evaluate the classification performance of the agent. The following section describes how performance of the classifier changes with individual news articles for all the participants, and how the model performs if words taught by all the participants are combined with the pre-trained Naive Bayes classifier.

4.5 Results

4.5.1 Indicators of crowd performance

We performed an analysis of variance (ANOVA) between the teachable agent's final F1 score, and each of the user attributes collected from pre-study questionnaire: gender, profession, interest, knowledge and prior teaching experience. Since the F1-scores were continuous and most of the user attributes were categorical in nature, we grouped the F-1 values using the categorical variables, measured the variance in each group and compared it with the overall variance of agent's performance. Participants' self-reported prior knowledge on the news categories was not observed to have any effect on the performance of the teachable agents ($p > 0.05$). Similarly, no significant relationship was observed between the other user attributes and final F1-score of the teachable agent (all $p > 0.05$). Therefore, pre-filtering of participants based on their responses in pre-study questionnaire was not performed for subsequent studies.

4.5.2 Teaching Efforts

Words taught during conversation

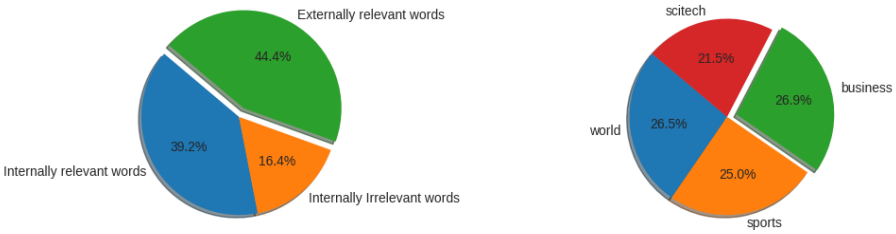
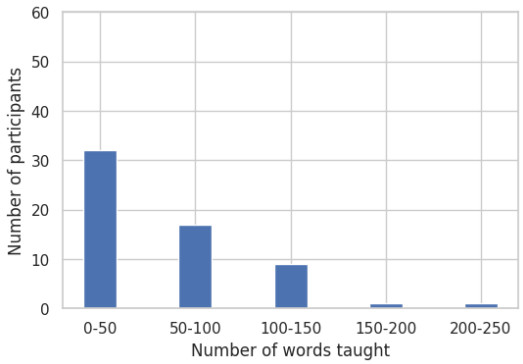
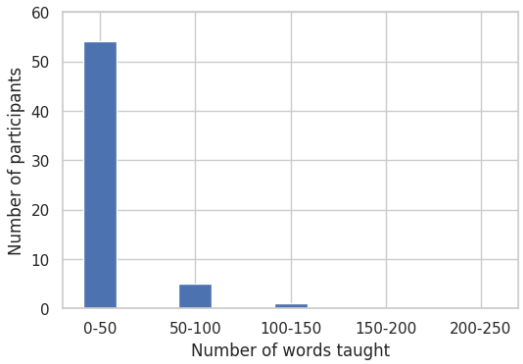


Figure 4.4: Proportion of words taught for each (a) type and (b) news category.

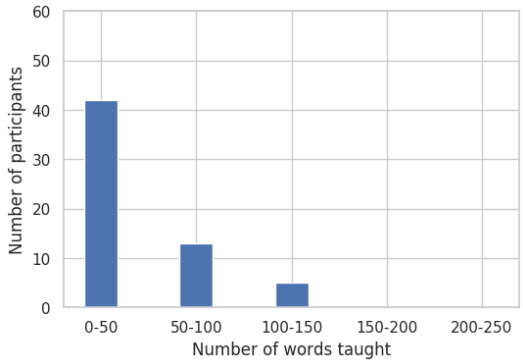
Each crowdworker individually taught an average of 146 words (SD = 155.08) during the conversation. High standard-deviation indicates that some of the participants taught too little and others taught a lot of words to the agent. Overall, crowdworkers taught a total of 8471 words across all interactions. Figure 4.4 illustrates the proportion of words taught for each type of word and different news categories. 44.4% of the overall taught words were not from the news article given in the interface but relevant to the topic in general. 39.2% of the words were relevant and also mentioned in the article, while the remaining 16.4% of the words taught were mentioned to be irrelevant to the topic being discussed. A category-wise frequency analysis revealed that conversations related to Business domain



(a) Internally Relevant



(b) Internally Irrelevant



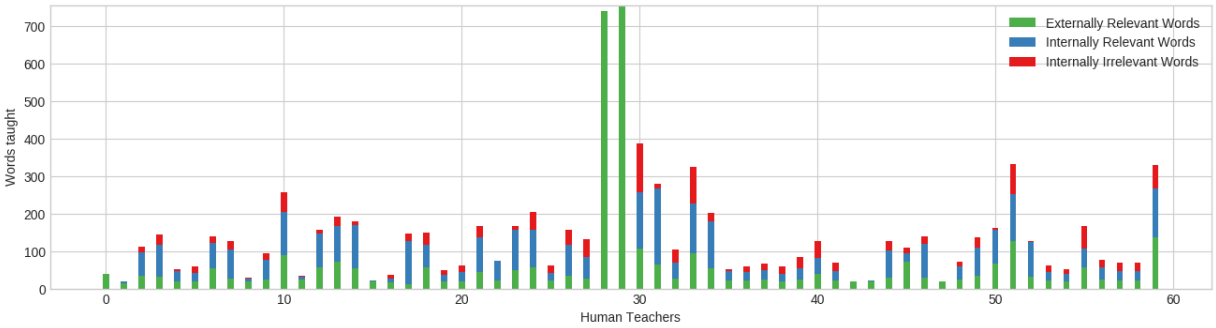
(c) Externally Relevant

Figure 4.5: Proportion of words taught by all the participants across (a) internally relevant (b) internally irrelevant, and (c) externally relevant words during the interaction.

contributed the most number of taught words (26.9%) while the conversations related to Science Technology contributed least number of words (21.5%). World and Sports contributed 26.5% and 25% words respectively that were taught during conversational interaction. Proportion of words taught by all the participants is shown in figure 4.5. Figure 4.6 shows the number of words taught by individual participant.

Average time spent

Crowdworkers spent an average time of 42.5 minutes for the entire experiment. No significant effect of time spent was noticed on the final F1-scores indicating classification



(a)

Figure 4.6: Words taught by individual crowdworkers during the interaction.

performance of the teachable agent ($p > 0.05$).

Number of dialogues exchanged

Average number of dialogues exchanged between crowdworkers and the teachable agent was 515 (SD=101). This metric was highly influenced by the number of words taught to the agent and the way they were exchanged. Some crowdworkers taught individual words in different dialogues, whereas some preferred teaching all the words in one message separated by spaces or commas. No significant effect of the number of dialogues was noticed on the final F1-scores indicating classification performance of the teachable agent ($p > 0.05$).

Number of times agent was tested

During the interaction, crowdworkers were allowed to switch between teaching and testing modes in order to validate the agent's performance. We monitored the number of times crowdworkers tested their agents while teaching in order to examine whether this has any effect on the agent's final classification performance. It was observed that the number of times a teachable agent was tested by the crowdworkers had no significant effect on its classification performance ($p > 0.05$).

4.5.3 Agent's Performance

Change in classification performance

We calculated the classification performance of the agent after each news article that was discussed during the conversational interaction. As mentioned before, although the classifier was trained online on the keywords captured from conversations on the current article, along with the keywords captured from all previous conversations, the performance was calculated "offline" on the entire test set of 7600 articles from the AG News Dataset treating individual article as an epoch. For this, we used the interactive variant of Multinomial Naive Bayes classifier as described in equation (4.3). Since the classifier was used without supervised pre-training, the initial performance was around 20% before the interaction. After the interaction, some of the most successful crowdworkers were able to increase the performance of the agent to around 70%, while for the least successful ones, the performance decreased to 10%. Results indicate that the final performance of classifier varied significantly across different participants. We did not find a direct co-relation between the number of words taught and the classification performance. This indicates that the quantity of the words captured alone does not impact the classifier's performance. Figure 4.7 shows the progression of F1-score with each article for 3 most successful and least successful teachers, that trained an interactive machine learner without supervised pre-training.

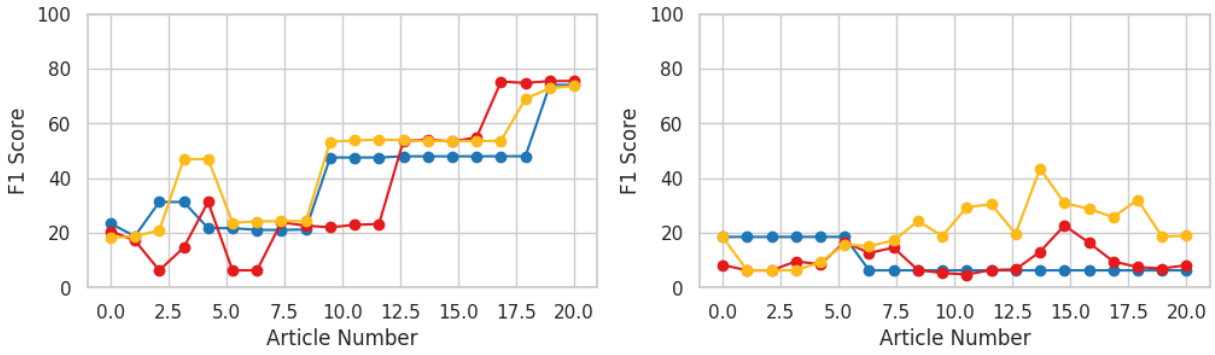


Figure 4.7: Change in accuracy of the agent when taught by 3 (a) most successful, (b) least successful crowdworkers, with no supervised pre-training of the interactive Naive Bayes classifier

Overall classification performance

Previous section describes how the performance of the classifier changes with each article discussed during the conversation without supervised pre-training. In this section, we describe the results of classifier's performance for other interactive variants of Naive Bayes with supervised pre-training as described in equation (3.11). These results were obtained "offline", by simulating the learning conditions after the experiment. Both statistical likelihood of words from relevant classes, and the user-defined likelihood obtained from conversations were used to calculate the posterior probability of test-documents. The classification performance of the interactive variants of Naive Bayes were compared with the two baselines for Bernoulli Naive Bayes (BNB) and Multinomial Naive Bayes (MNB) respectively. The comparison was made between most successful, least successful, and combination of all crowdworkers who taught the teachable agent during the experiment. Precision, recall and F1 scores for the interactive variants are described in Table 4.1

Model	**Precision**	**Recall**	**F1-Score**
Without Teachers (Baseline)			
Bernoulli Naive Bayes	0.8626	0.8584	0.8593
Multinomial Naive Bayes	0.8899	0.8902	0.8900
Best Teacher			
Interactive Bernoulli Naive Bayes	0.8658	0.8672	0.8664
Interactive Multinomial Naive Bayes	**0.8972**	**0.9042**	**0.9006**
Worst Teacher			
Interactive Bernoulli Naive Bayes	0.8145	0.8247	0.8196
Interactive Multinomial Naive Bayes	0.8729	0.8709	0.8719
All Teachers			
Interactive Bernoulli Naive Bayes	0.8532	0.8578	0.8558
Interactive Multinomial Naive Bayes	0.8847	0.8830	0.8838

Table 4.1: Comparison of baseline classifiers with interactive variants of Naive Bayes with supervised pre-training, for best teacher, worst teacher and all teachers.

4.5.4 Post-study questionnaire

After the experiment, participants were asked to fill a post-study questionnaire designed for the assessment of perceived usability of a system. This questionnaire was adapted from the IBM Computer System Usability Questionnaire (CSUQ) [83]. CSUQ is a 5-point

usability scale questionnaire, similar to the 18-item version of Post-Study System Usability Questionnaire (PSSUQ) [82], with slight changes to the wording due to the change in research context. We used the 16-item version of CSUQ that produces four scores: three well defined sub-scales and an overall measurement for the perceived usability of a system.

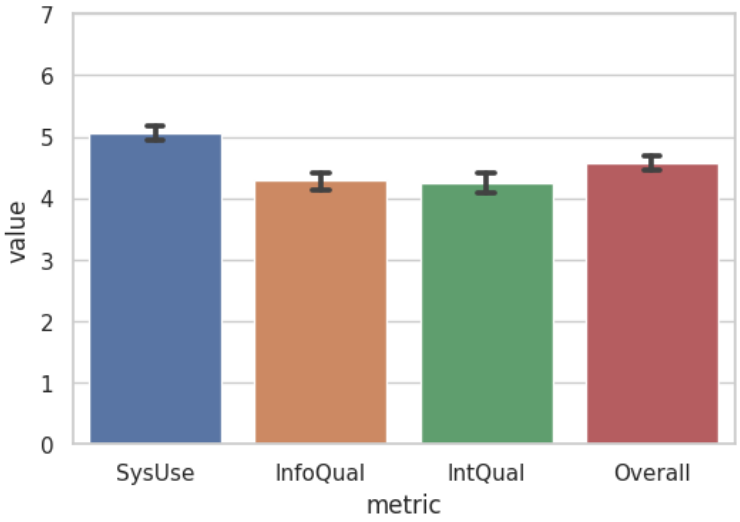


Figure 4.8: Median scores for usefulness, information quality, interface quality and overall usability of the system from CSUQ

Median scores within the 95% confidence intervals for overall perceived usability along with three sub-scales are shown in Figure 4.8. Mean rating for System Usefulness (SysUse) was recorded as 5.14 (SD=0.36). For Information Quality (InfoQual) and Interface Quality (IntQual), participants reported an average score of 4.28 (SD=0.48) and 4.2 (SD=0.53) respectively. Overall, the mean perceived usability of the system was recorded to be 4.59 (SD=0.29).

4.6 Discussion

A major aspect of this study was to investigate whether crowdworkers can directly interact with a teachable conversational agent and interactively train the underlying machine learning classifier. This is different from the traditional approach where crowdworkers only provide the labels for training data, which is later used to train the classifier offline. Results from the experiment reveal that sincere crowdworkers who are good at teaching can

iteratively improve classifier's performance with even limited conversational interactions. However, ineffective teaching may result in sub-optimal performance of the underlying classifier as illustrated in Figure 4.7. The difference between effective and ineffective teaching cannot solely be quantified by the number of words taught during conversational interaction and may also depend on the quality of teaching as well. Identifying the quality of teaching from conversational interactions remains an open challenge in this area and may need further investigations. Similar to the situation where interactive learning is used without supervised pre-training, the results also indicate that the performance of the classifier improves with most effective teachers and degrades for least effective teachers when used with supervised pre-training as described in Table 4.1. An interesting finding is that the combined effect of teaching from all the crowdworkers may actually reduce the overall performance of the classifier in an interactive setting. This implies that interactive learners who aim to learn from conversational interactions should be used individually with different human-teachers rather than directly learning from a group of teachers. Learning from a lot of sources may affect the performance of the learner if the proportion of ineffective teachers is significantly more than effective ones, and effective and ineffective sources are indistinguishable. Finally, results from the post-study questionnaire indicate that teaching through conversational interactions is perceived well by the participants in terms of usability of the system and provide a meaningful engagement experience.

Chapter 5

Experiment 2: Learning By Teaching

The goal of this study is to investigate the effectiveness of the learning-by-teaching paradigm within the context of crowdsourcing tasks. One part of this study is focused on facilitating crowdworkers in teaching a classification task to an AI agent. The other part is concerned with simply providing them more instructions to do the task. Specifically, we are interested in knowing whether crowdworkers can improve their own performance by teaching the task to a virtual AI agent, compared to a situation where they do the same task themselves with additional task instructions. We compare the pre- and post-interaction performance of crowdworkers in the two conditions to validate the effectiveness of learning-by-teaching technique. Additionally, we also capture participants' opinion on the usefulness and perceived enjoyment during the task through Activity Perception Questionnaire from the Intrinsic Motivation Inventory. Thus, this study serves as an important task of informing the usefulness of Teachable Agents for humans, especially in the context of crowdsourcing.

5.1 Design

The experiment used a between-subject design with the "task interaction technique" in training phase as an independent variable. We chose two interaction techniques to seek information from the crowdworkers. In the control condition, crowdworkers self-classified the news articles with additional task-instructions corresponding to the rubric. These instructions were displayed on a panel alongside the text to classify. In treatment condition, crowdworkers were asked to teach the classification to Kai: the virtual AI agent embedded within Curiosity Notebook. The agent used conversational interventions to deliver the task instructions and elicit teaching from crowdworkers. The agent acted as a less intelligent

entity with the desire to learn the classification task from crowdworkers. Both experimental conditions conveyed the same amount of instructions for the task as defined by the rubrics. In the control condition, crowdworkers were treated as annotators, whereas in the treatment condition, they were treated as human-teachers. The entire experiment was divided into 3 parts with first and third part measuring pre-interaction and post-interaction baselines and second part representing the experimental condition. Dependent measures from the study are as follows:

- **Words Captured**, number of words captured from each article for each condition.
- **Task Completion Time**, the duration of time from the beginning to completion for pre-interaction and post-interaction baseline tasks.
- **Participants' Accuracy**, the proportion of articles labelled correctly in pre-interaction and post-interaction baseline tasks.

We also measured the following dependent variables from the Activity Perception Questionnaire in Intrinsic Motivation Inventory:

- **Interest/Enjoyment**, self-reported enjoyment of the activity from the participants in respective condition.
- **Value/Usefulness**, self-reported usefulness of the activity from the participants in respective condition.

Participants were given 8 articles in first and third part respectively, and 4 articles in the second part that represented one of the experimental conditions. In total, each participant saw 20 articles during the experiment.

5.2 Participants

We recruited 100 crowdworkers from Amazon Mechanical Turk (38 females, 62 males), 22 to 65 years old (M= 33.74, SD= 9.24). Participant pool represented a variety of professions including freelancers (42), managers (23), engineers (15), home-makers (8), and designers (3). Remaining 9 participants were self-employed. 94% of the participants were native English speakers, but all reported some prior experience with conversational agents on a 7-point Likert scale (M=5.58, SD=1.56). 37 % of the participants reported prior experience

in teaching a classification to someone else, the other half had no prior experience on teaching (63%). Regarding the prior knowledge on the 4 given news categories, participants rated most for SciTech (M=5.26, SD=1.43), followed by Sports (M=5.09, SD=1.71), World (M=5.01, SD=1.45) and Business (M=4.08, SD=1.52).

The HITs were posted with the title "Teach How to Classify News Articles to a Chatbot". Participants received $0.5 USD for the pre-study questionnaire on demographics, $2 for both baseline tasks, $2 for completing the condition task (teaching the classification or self-classification with instructions), and $0.5 USD for the post-study questionnaire. The two baseline tasks took approximately 5 minutes, interaction phase took approximately 10 minutes and the pre- and post-study questionnaires took 2-5 minutes each to complete. As in the first experiment, this study was also conducted within Curiosity Notebook. Only participants using Chrome and Firefox were allowed to participate in order to reduce the possibility of browser incompatibility.

5.3 Procedure

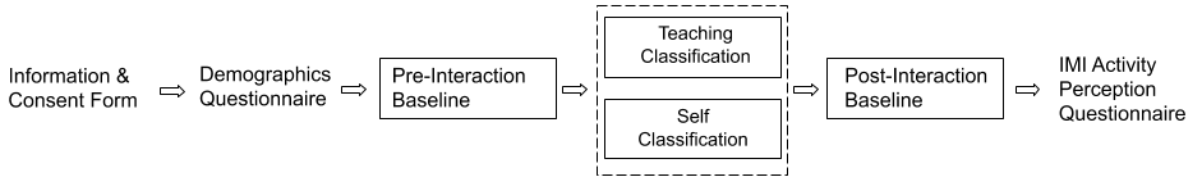


Figure 5.1: Study procedure for experiment 2

Crowdworkers were first given a series of text-classification tasks to capture their pre-interaction baseline performance. Then, they were placed in independent conditions and told that the purpose of this part is to help them more accurately annotate the articles for text-classification. In this phase they were asked to mark certain words from the text that helped them choose a specific category for the overall article. Workers were divided into two experimental conditions. In control condition, they were given a set of rubric-instructions to follow while annotating the text to classify. These rubric-instructions asked them to further specify which words in the text were most and least relevant to the category belonging to the article. In treatment condition, workers interacted with Kai, a conversational agent that asked them to teach the text classification task. Kai used teaching-guidance to elicit the same amount of information as gathered in control condition. Both rubric-instructions and teaching-guidance were different forms of the same underlying

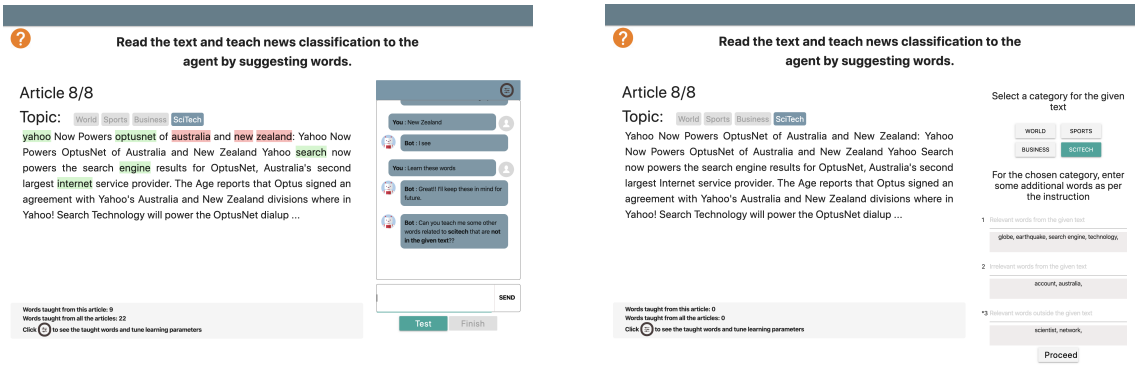
(a) Pre-interaction (b) Post-interaction

Figure 5.2: Interface for (a) pre-interaction, and (b) post-interaction task in experiment 2

heuristics as described in subsection 3.3.3. These heuristics were carefully designed to optimize the performance of Naive Bayes classifier that was ultimately being trained from Turkers responses. We ensured that the teaching-guidance used in the treatment group and the rubric-instructions shown in the control group convey the same amount of information. Effectively, workers were exposed to the same type and amount of information, but they assumed different responsibilities during the training process. Crowdworkers in control condition were supposed to do the task themselves, whereas crowdworkers in the treatment condition were supposed to teach the task to a virtual AI agent. Finally, participants from both conditions were asked to perform a post-interaction baseline task, in order for us to evaluate whether there is any change in their performance. Study procedure for the experiment is described in Figure 5.1

5.4 Task Interface

As described above, the task was divided into three stages corresponding to pre-interaction, experimental condition, and post-interaction task. In each of the stages, crowdworkers were supposed to identify the correct class for a news-snippet sampled from the AG news dataset. Pre-interaction and post-interaction stages were identical and asked the crowdworkers to select a correct document class belonging to the given text. After this, they could click the next button to proceed to the next article. Figure 5.2 describes the task interface in pre-interaction and post-interaction task.



(a) Interface for Teaching Classification

(b) Interface for Self Classification

Figure 5.3: Study conditions in experiment 2 with interfaces for (a) teaching-classification, and (b) self-classification

After the pre-interaction task, crowdworkers were redirected to the second part that corresponded to the experimental condition. In this part, they were randomly assigned to one of the two experimental conditions. In both the conditions, participants were asked to identify the document class for the given news articles. Further, they were asked to provide some words that helped them make the decision while selecting the class. The control condition presented a passive textual interface containing instructions for the participants. In treatment condition, crowdworkers interacted with a teachable conversational agent that expressed an eagerness to learn text-classification. The agent used teaching guidance as described in subsection 3.3.3, to ask the participants about specific words related to the topic. The teaching guidance was exchanged in conversational form and conveyed the same amount of information as it was conveyed in the control condition. In both conditions, new words were provided by typing them in a textbox. Once the words were taught, participants could ask the agent to learn them. Like in the previous experiment, participants in this condition could also test the agent's classification accuracy by switching to the testing mode as described in Figure 4.3. The task interface in two experimental conditions is described in Figure 5.3. After the second part, participants were redirected to the post-interaction task to label few more news articles without explicit cues from the interface.

5.5 Analysis Methods

In this section, we describe the metrics and methods that were used to compare the teaching condition with self-classification. As mentioned above, we considered words captured during the interaction phase, and change in performance in pre-interaction and post-interaction parts as the primary performance metrics for each experimental condition. The performance in pre/post-interaction baselines was defined as the average time spent, and average classification accuracy of the crowdworkers. We were interested in knowing whether there is any effect of the interaction technique on workers' performance. Specifically, we wanted to know whether teaching a task to a virtual agent makes crowdworkers better, compared to a situation when they do the task on their own. We also analyzed the perceived usefulness and interest that the participants reported in the post-study questionnaire.

5.5.1 Words Captured From Interaction

We calculated the total number of words captured during the interaction in each experimental condition. Since this data was discrete and non-normal, we performed Kruskal-Wallis test to examine the effect of interaction techniques on the number of words captured.

5.5.2 Pre- and Post-Interaction Performance

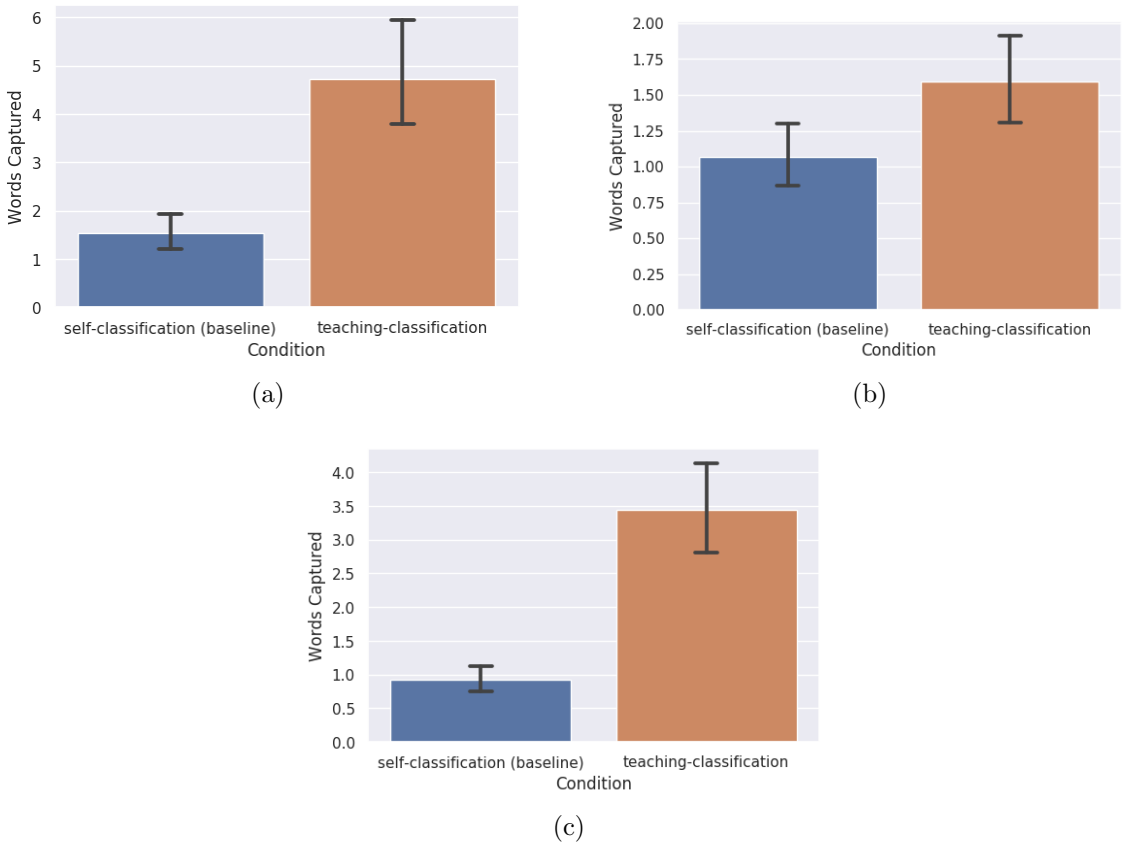
As described above, we calculated participants' accuracy and task completion time between pre-interaction and post-interaction baselines to analyze crowdworkers' performance in each condition. We compared the time data in two ways: across the post-interaction baselines for both groups, and individual difference between pre-interaction and post-interaction baselines. Since the time data was continuous and satisfied normality after log transformation, we performed one-way ANOVA and pairwise t-tests with Bonferroni correction, to compare the corresponding distributions.

5.5.3 Post-study Questionnaire

After the experiment, participants were asked to fill a post-study questionnaire to report their opinion on the perceived enjoyment and perceived usefulness based on the condition they were exposed to, during the task. These ratings were recorded on a 7-point scale and analyzed using Kruskal-Wallis test, followed by Mann-Whitney test for post hoc analysis.

5.6 Results

5.6.1 Words Captured From Interaction



(a) (b)

(c)

Figure 5.4: Words captured during the interaction phase in experiment 2

Figure 5.4 illustrates the average number of words captured in each condition. In the control condition, participants were found to contribute an average of 1.55 words (SD=1.27) that were present in the text and also relevant to the chosen category (internally relevant), 1.07 words (SD=0.81) that were present in the words but not relevant to the chosen category (internally irrelevant), and 0.92 words (SD=0.68) that were not present in the text but relevant to the category. In the treatment condition containing a teachable conversational agent, participants contributed an average of 4.72 internally relevant words (SD=3.96), 1.59 internally irrelevant words (SD=1.12), and 3.44 words that were relevant

but not present in the news-snippet (SD=2.56). We excluded stopwords, and common padding words while counting, as they may have appeared in treatment condition owing to the conversational nature of the interface. In total, participants provided an average of 3.53 words (SD=2.49) in the control condition involving self-classification of news articles, and 9.74 words (SD=6.11) in the treatment condition containing a teachable conversational agent. Since the data exhibited non-normality, we performed Kruskal-Wallis test for significance testing and Mann-Whitney test for post hoc analysis. These are the non-parametric equivalent of ANOVA and unpaired t-test respectively. Treatment condition the teachable conversational agent was found to capture significantly more words than the control condition involving self-classification ($p < 0.05$). This indicates that teaching the classification is more beneficial than self-classification with passive instructions for both human-teachers and machine-learners .

5.6.2 Pre- and Post-Interaction Performance

Average Time Spent

In control condition with annotation interface, participants were found to complete the pre-interaction task in 91.96 seconds (SD=109.9), and post-interaction task in 79.68 seconds (SD=198.64). In the treatment condition with teachable conversational agent, participants completed the pre-interaction task in 100.48 seconds (SD=78.10), whereas the post-interaction task was completed in 67.98 seconds (SD=54.87). Visual inspection of the task completion times suggested non-normality, which was confirmed by Shapiro-Wilk and Anderson-Darling tests. To compensate this, we applied log-transformation on all data points for task completion time. Log transformations were only applied for statistical tests and all times presented in the thesis are actual measured values. After this, we performed one-way ANOVA to investigate if the participants actually got faster after respective conditions. No significant effect was observed in task completion time between pre-interaction and post-interaction tasks for the control condition($p > 0.05$). For treatment condition with teachable conversational agent, the test showed a significant improvement in task completion time during post-interaction task ($p < 0.05$). Finally, we compared the task completion time across both the conditions. No significant difference was observed between pre-interaction, or post-interaction times across the conditions ($p > 0.05$). Figure 5.4 illustrates the average number of words captured in each condition.

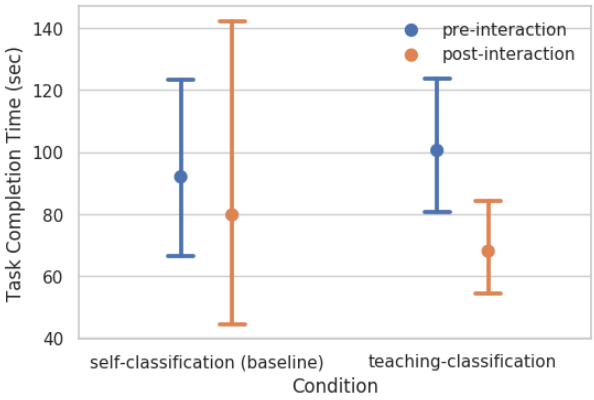


Figure 5.5: Average time taken to complete pre-interaction and post-interaction tasks across both conditions

Classification Accuracy

We calculated the average accuracy of participants in labelling the news articles during pre-interaction and post-interaction tasks. Accuracy of the participants was calculated by following equation:

$$Accuracy = \frac{Number\ of\ articles\ correctly\ labelled}{Total\ number\ of\ articles} \times 100 \tag{5.1}$$

The average accuracy of participants in pre-interaction task for the control condition was 78.5% (SD=18.21). For treatment condition, the pre-interaction accuracy was 77.75% (SD=17.73). In both conditions, the accuracy was observed to slightly increase in post-interaction tasks (79.5% in control vs 81% in treatment). Similar to time data, accuracy of the participants in pre- and post-interaction satisfied normality after log-transformation. We performed one-way ANOVA with interaction technique as the independent variable to investigate its effect on participants' performance. The increase in accuracy in both conditions was not found to be significant ($p > 0.05$). Figure 5.6 describes the average accuracy of crowdworkers in pre-interaction and post-interaction tasks across the two conditions.

5.6.3 Post-study Questionnaire

After the experiment, participants were asked to fill a post-study questionnaire on Activity Perception from Intrinsic Motivation Inventory (IMI). The IMI activity perception

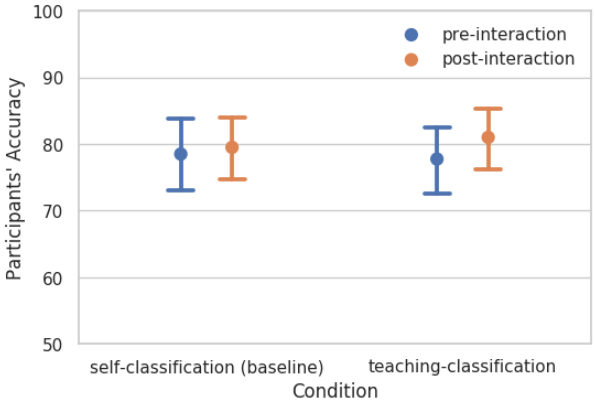
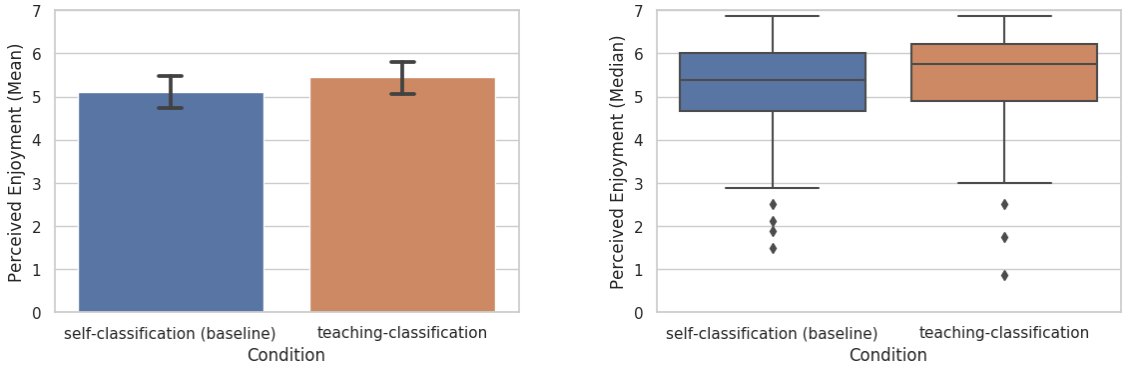


Figure 5.6: Classification accuracy of crowdworkers in pre-interaction and post-interaction tasks across both conditions.

questionnaire is a collection of 25 questions containing 7-point Likert scale type questions. We selected a subset of the scale containing 17 questions that measured participants' interest/enjoyment in the activity as well as the perceived value/usefulness. Overall, we collected a total of 100 survey responses corresponding to 50 participants in each experiment condition.

Interest/Enjoyment

The experimental condition with the teachable conversational agent was rated higher (M=5.45, SD=1.31) on the scale of enjoyment than the condition with passive rubric instructions (M=5.1, SD=1.39). Since the survey data was ordinal and assumed non-normality, we performed Kruskal-Wallis test for significance testing. The perceived enjoyment rating between the two condition was not found to be significant ($p > 0.05$). Figure 5.7(a) illustrates the mean enjoyment ratings for both conditions. Figure 5.7(b) describes the median rating within first and third quartile range.



(a) Mean Score

(b) Median Score

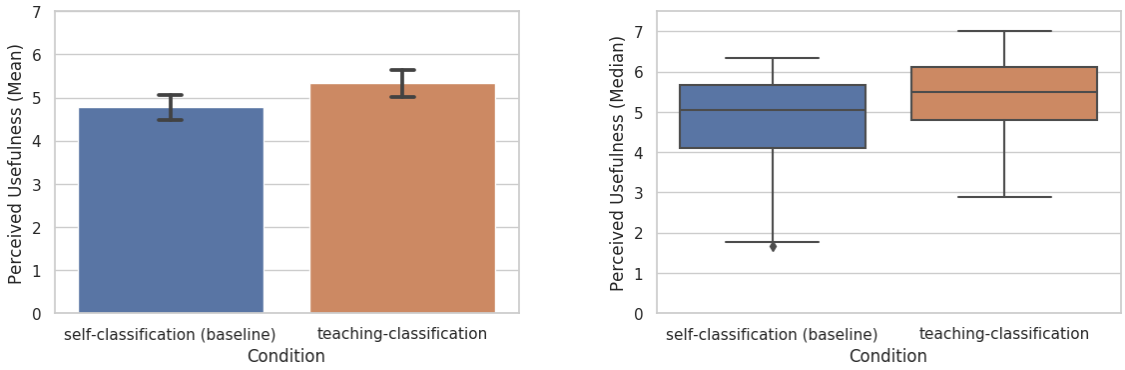
Figure 5.7: IMI enjoyment

Value/Usefulness

We examined the average ratings for perceived usefulness of the activity that was performed by the crowdworkers in each experimental condition. Mean score recorded for the control condition with teachable conversational agent was 4.76 (SD=1.11). For treatment condition, the average usefulness score was recorded to be 5.34 (SD=1.09). We performed Kruskal-Wallis test with mean usefulness rating as the dependent variable as the data was ordinal and assumed non-normality. Results indicate that participants in treatment condition considered the interaction with teachable agent more valuable than the participants in control condition ($p < 0.05$). Figure 5.8(a) illustrates the mean enjoyment ratings for both conditions. Figure 5.8(b) describes the median rating within first and third quartile range.

5.7 Discussion

The results from this experiment provide some interesting evidence that favours the use of teaching over self-classification in the context of crowdsourcing studies. It was found that letting crowdworkers teach a classification to an agent captures more words from them compared to the condition where crowdworkers self-classify the articles as illustrated in Figure 5.4. Thus, teaching a task can provide better results while eliciting information from the human-teachers in an interactive setting. Further, comparing the pre-interaction and post-interaction performance of the crowdworkers in both conditions reveal that participants in



(a) Mean Score

(b) Median Score

Figure 5.8: IMI usefulness

the treatment condition with responsibility to teach news classification showed significant improvement in task completion time and participants' accuracy during post-interaction phase compared to the participants in control condition who self-classified news articles (Figure 5.5 and Figure 5.6 respectively). These results indicate towards the presence of Protégé Effect that implies that teaching, pretending to teach, or preparing to teach an information to someone else (eg. a Teachable Agent) ultimately helps the teacher learn that information. However, more studies should be conducted to examine the actual learning that comes from long-term and short-term interactions with the Teachable Agent. Finally, the IMI usefulness scores reported in post-study questionnaire shows that the teaching was perceived to be more useful and valuable than the self-work (Figure 5.8). This suggests that teaching is an effective method for eliciting information from crowdworkers, and using a teachable conversational agent in the crowdsourcing context can improve workers performance in terms of accuracy and task completion time.

Chapter 6

Experiment 3: Dynamics of Trust

In this experiment, we investigate whether crowdworkers would trust an AI agent that they themselves taught. Specifically, we are interested to determine if Turkers would prefer incorporating teachable agents in their workflow if they diligently teach them for certain human intelligence tasks, and what factors might matter in influencing trust in either a positive or negative way.

6.1 Design

The experiment used an observational study design with the preference over task delegation as a dependent variable. Total experiment was spanned across two parts: first part pertaining to the teaching phase, and an optional bonus phase to label extra news-snippets.

6.2 Participants

We recruited 40 crowdworkers from Amazon Mechanical Turk (18 females, 22 males), 21 to 54 years old (M= 29.97, SD= 9.38). Participant pool represented a variety of professions including managers (10), IT technicians (10), engineers (6), teachers (5), nurse (3) and designers (3). Remaining 3 were self employed. 89% of the participants were native English speakers, but all reported some prior experience with conversational agents on a 7-point Likert scale (M=5.63, SD=1.31). 30 % of the participants reported prior experience in teaching a classification to someone else, the other half had no prior experience on

teaching (70%). Regarding the prior knowledge on the 4 given news categories, participants rated most for SciTech (M=5.38, SD=1.43), followed by World (M=5.2, SD=1.14), Sports (M=4.78, SD=1.76) and Business (M=4.48, SD=1.57).

The HITs were posted with the same title as previous two experiment, "Teach How to Classify News Articles to a Chatbot". Crowdworkers who participated in the previous two experiment were not allowed to participate in this study. Eligible participants received $0.5 USD for the pre-study questionnaire on demographics, $2 for the teaching task, and $0.5 USD for the post-study questionnaire. Further, for every correctly labelled sample in the bonus task, they received an amount of $0.17 USD. The maximum possible payment for the bonus task was $2 USD (12 articles * $0.17 USD per correctly labelled article). The teaching task took approximately 10 minutes, bonus took around 0-5 minutes depending on whether the crowdworker decide to delegate the task or do it all by themselves. Pre- and post-study questionnaires took 2-5 minutes each to complete. As in the first two experiments, this study was also conducted within Curiosity Notebook. Only participants using Chrome and Firefox were allowed to participate in order to reduce the possibility of browser incompatibility.

6.3 Procedure

We recruited a new set of workers on Amazon Mechanical Turk. Crowdworkers were first given a series of 8 text classification tasks and asked to teach a virtual teachable agent named Kai that delivered the same conversational interventions as described in Experiment One. During this phase, workers were told that their future compensation may depend on how successfully they teach Kai during the task. In this part, they were also expected to validate the classification performance by observing the agent's accuracy in test mode, similar to the first experiment. In the second part, they were presented with a bonus task to label 12 more news-snippets. The bonus task was optional and gave them an option to either do the task themselves, delegate it to an agent. Crowdworkers were asked to choose one of the two available options, or skip the bonus task altogether. For the delegation part, they could choose a value on a 12-point slider to decide which portion of task they want to do themselves and which portion they want to delegate to the agent they recently taught. For either of the options, they were also asked to provide a reason for their choice in an open-form text box. After the bonus task, they were asked to fill a questionnaire, which investigated their level of trust on the agent and factors that influence their trust level.

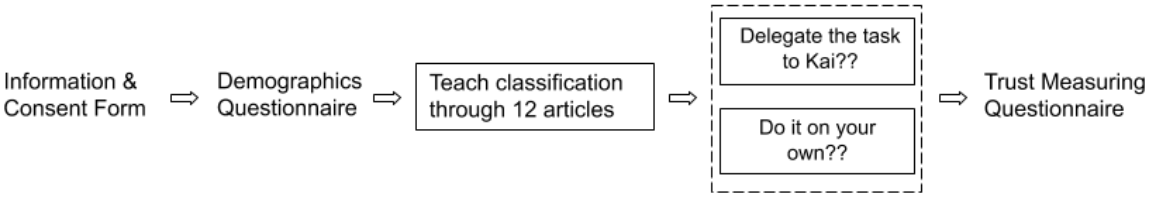
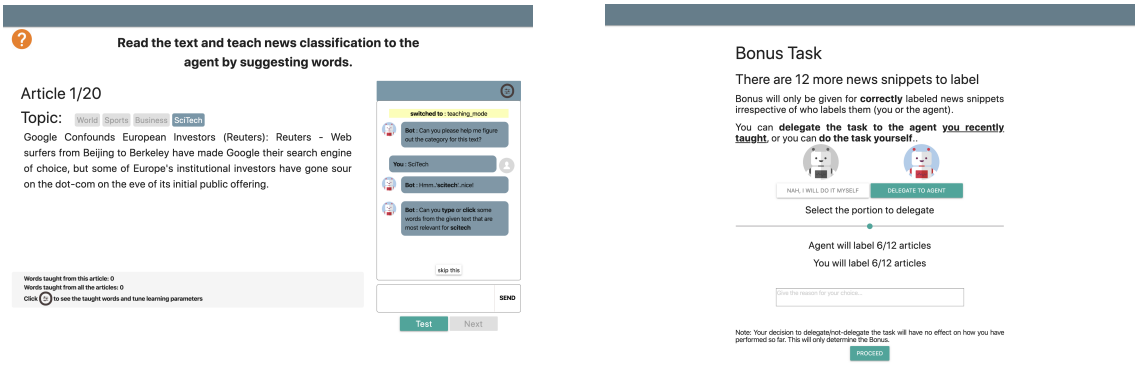


Figure 6.1: Study procedure for experiment 3

6.4 Task Interface

As described above, the entire experiment was divided into two parts. The first part was about teaching the agent and similar to the teaching parts described in previous experiments. Participants were supposed to teach a total of 8 articles to Kai: the teachable conversational agent embedded within the experimental interface. Like previous experiments, participants in this experiment could also evaluate the performance of their agent by switching to the testing mode and observing its classification performance on unseen articles. Figure 6.2(a) describes the task interface in teaching mode. After the teaching part, participants were provided with a bonus task where they could earn more by labelling some additional news snippets. They were also informed that the additional payment for bonus task will be calculated only based on the number of correctly labelled samples.



(a) Teaching mode

(b) Delegation mode

Figure 6.2: Task interface for experiment 3 while (a) teaching the agent, and (b) delegating the task.

Based on the information provided in the bonus task, participants were allowed to either delegate a portion of the task to the agent and do the rest themselves, or do the entire task on their own without delegating, knowing that the bonus will be only paid for correctly labelled articles. For this, the participants were provided with a 12-point slider in order to select the portion of task to delegate. Figure 6.2(b) describes the task interface for delegation mode.

6.4.1 Agent's Learning

Unlike previous experiments, where the agent was purely learning from the conversational interaction, agent in this experiment was only simulating its learning based on the number of news articles covered and words taught from each article. The reason behind this change was to ensure that all the agents are equally accurate in classifying news articles after the teaching part. This was important because the experiment was designed to measure the trust of participants while delegating the tasks that involve monetary compensation. Since a part of this decision may come from their perceived accuracy of the agent, it was important to account for teachers who may not succeed in teaching, resulting in less-accurate agents. As it is difficult to successfully separate ineffective teachers from effective ones in the beginning of the task, we decided to simulate agent's learning instead of letting it learn online for more consistent results towards task delegation.

6.5 Analysis Methods

In this section, we describe the metrics that were used to investigate the dynamics of trust relating to the interaction around a teachable agent. From the observational study that involved teaching a task to an agent, we examined whether participants would later delegate similar tasks involving monetary compensation to the agents that they recently taught, or would they rather do the tasks themselves.

6.5.1 Portion of Tasks Delegated

We calculated the number of times participants decided to delegate a portion of the task to the agent that they taught. This was used for descriptive analysis on the proportion of overall tasks performed by the agent, and the participant.

6.5.2 Post-study Questionnaire

After the experiment, participants were asked to fill a post-study questionnaire to report their opinion on general self-efficacy, task specific competence and general trust towards the automation offered by the system containing a teachable agent.

6.6 Results

6.6.1 Portion of Tasks Delegated

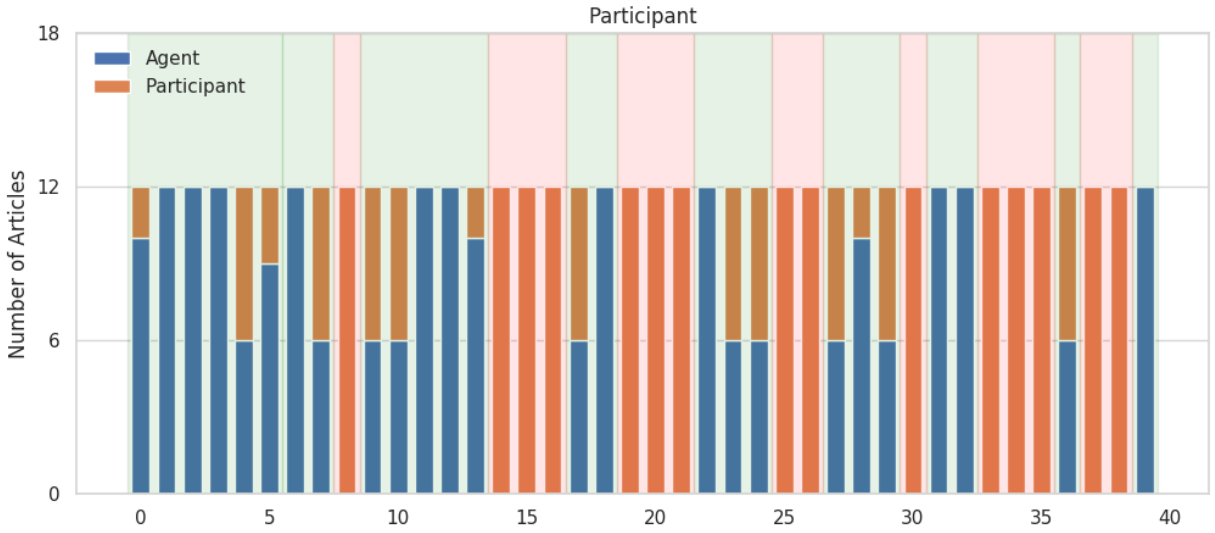


Figure 6.3: Tasks delegated to the agent

Each participant was given a total of 12 articles in the bonus task, and a choice to either label all the articles themselves, or delegate a portion of those to the agent they recently taught. It was clearly mentioned that only the correctly labelled articles in the bonus task will be considered while calculating the bonus amount. 62.5% (N = 25/40) of the participants trusted the agent and delegated a portion of the bonus task. The remaining 37.5% (N = 15/40) decided to trust their own skills to classify news articles in order to maximize the bonus amount. Among the participants who delegated the tasks to the agent, 44% (N = 11/25) decided to assign all 12 articles to the agent, while 40%(N = 10/25) assigned half the articles (6/12) in order to compare agent's work with their own.

Figure 6.3 represents the portion of tasks completed by the agent and the participants. Regions with green portion denote the instances where participants trusted the agent, whereas regions with red portion denote the instances when participants did not trust the agent and did all the articles in the bonus task themselves. Note that participants either did not trust the agent at all, or trusted them at least as much as they trusted their own skills in correctly classifying the news articles.

6.6.2 Post Study Questionnaire

Participants completed a post-study questionnaire after the experiment that was designed for the assessment of trust and competence and self-efficacy. The questionnaire was adopted from a combination of three surveys focused on general self-efficacy [68], task-specific competence [39], and empirically determined scale of trust between people and automation [69]. A total of 40 survey responses were gathered and analyzed using Kruskal-Wallis test with the delegation decision as an independent measure.

General Self Efficacy

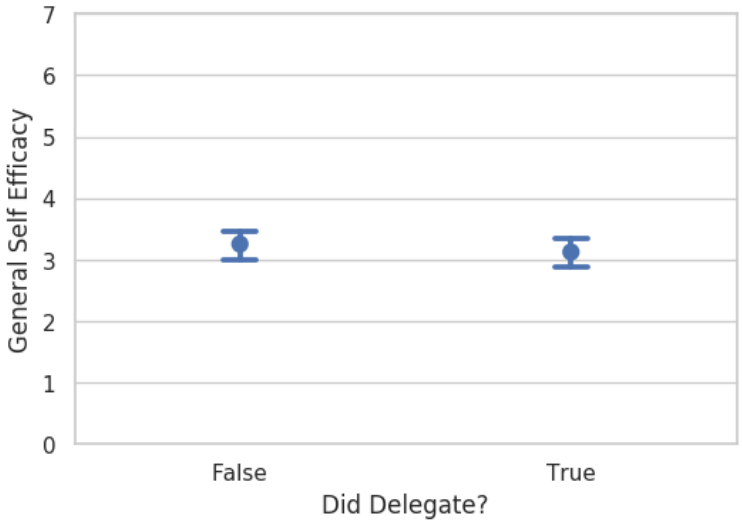


Figure 6.4: General self efficacy

Figure 6.4 represents the average self-efficacy reported by the participants on a 4-point Likert scale. Participants who delegated the task to the agent in the experiment reported

lower self-efficacy (M=3.12, SD=0.61) than the participants who did not delegate the task and labelled all the articles themselves (M=3.25, SD=0.47). However, no statistical significance was observed between the two groups ($p > 0.05$).

Task Specific Competence

Participants submitted their perceived competence on a 7-point scale adopted from the Intrinsic Motivational Inventory. Similar to general self-efficacy, average task competence reported by the participants who did not delegate the task to the agent was found to be higher (M=5.73, SD=0.76) than those who delegated (M=5.54, SD=0.92). This indicate that participants who were confident in their own skills to perform the news snippet classification task, were less likely to delegate it to the agent they taught. Likewise, participants who delegated a portion of the task were slightly less confident in their own competence to correctly label all the articles. In order to confirm the effect of competence for the task on the decision to delegate, we performed Kruskal-Wallis test with perceived competence rating as a dependent variable. However, the difference in scores reported was not found to be significant between the two groups ($p > 0.05$). Figure 6.5 represents the mean scores on perceived competence as reported by participants in the task.

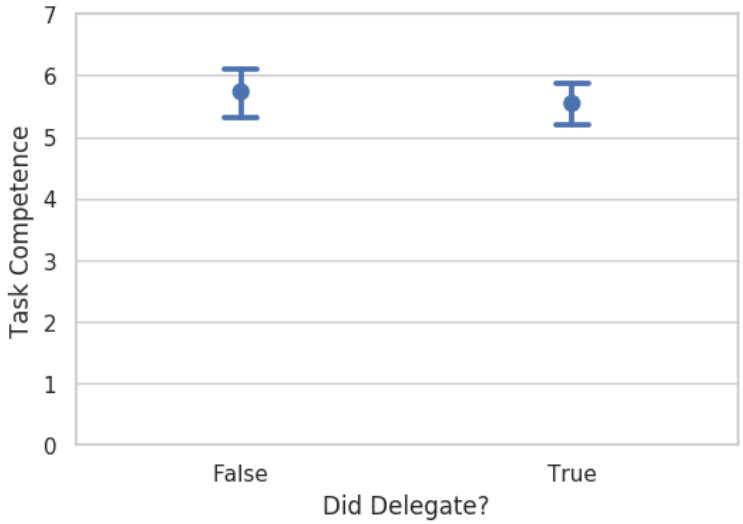


Figure 6.5: Task specific competence

Trust Towards AI

We examined the average ratings for self-reported trust of participants on a 7-point scale for systems containing automation through agents. Mean score reported by the participants who did not delegate the task to the agent was 3.89 (SD=0.82). On the other hand, participants who delegated a portion of the task to the agent reported a mean trust score of 5.16 (SD=0.84). Kruskal-Wallis test confirmed that that difference in trust ratings is highly significant between the two groups of participants ($p < 0.01$). This implies that using the teachable agent metaphor to perform a task may not be sufficient to gain people's trust. However, more studies should be performed to analyze the dynamics of trust with teachable agents, and agents that do not involve humans in their learning. Figure 6.6 represents the average trust rating towards the system with teachable conversational agent as provided by the participants.

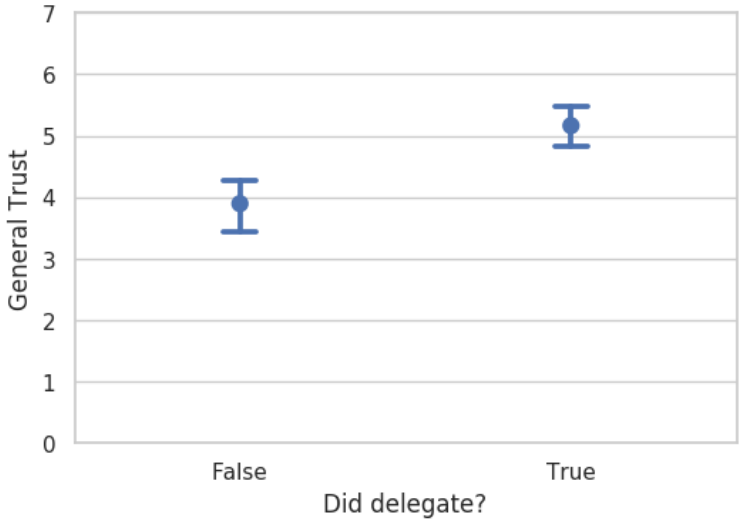


Figure 6.6: General trust on AI systems

6.7 Discussion

The primary purpose of this experiment was to examine whether the crowdworkers embrace the concept of teachable conversational agent, and adopt them in their workflow for the tasks they were originally taught on. Our findings reveal that while majority of the

crowdworkers preferred delegating a portion of their tasks to the agent they themselves taught, a significant portion of crowdworkers decided to do the entire task themselves. Further, participants who delegated the task assigned at least 50% of the work to the agent and kept lesser work for themselves. An interesting result was found in cases where participants delegated exactly half of the work to the agent in order to compare its accuracy with their own performance. These results are illustrated in Figure 6.3. Through the post-study questionnaire, it was observed that participants who did not delegate the task to their agent reported less trust on automated systems and were more confident in themselves and their own ability to do the same task as illustrated in Figures 6.6, 6.4 and 6.5 respectively. Similarly, participants who delegated the task reported lesser self-efficacy and task-specific competence compared to those who did not delegate. Our findings suggest that the dynamics of trust towards teachable agents still depend significantly on how people perceive automated systems in general. More experiments should be conducted to investigate if these results vary if a teachable agent taught by other people is presented for task delegation.

Chapter 7

Conclusion

In this thesis, we describe the domain of teachable-agents, and how that can be used to view an interactive machine learning system as a teacher-learner interactivity. We contributed to this area by focusing on the interaction from both teaching as well as learning perspective, and touching the dynamics of trust within human-teachers and agent-learners.

In chapter 2, we carried out a literature review related to previous work on interactive machine learning, conversational agents, and agent-based interactions within the human-computer-interaction research. We found that the theme of treating humans as teachers, and not just annotators, has been discussed in both machine learning and HCI literature. However, most of the existing work lacks a formal discussion over human-agent interaction and demands stronger collaboration between the two research communities. Informed by the previous work, we introduced the concept of teachable conversational agents, that leverages teaching guidance in the form of conversational cues to elicit better responses from human teachers and learn a classification task.

In chapter 3, we present an interactive variant of Naive Bayes classifier that relaxes the 'naive' independence assumption of features and utilizes additional features captured from conversational interactions to modify the posterior probabilities while making predictions.

In chapter 4, we examine the effectiveness of our interactive classification algorithm and describe how its performance may vary based on the quality of instructions delivered by human teachers.

In chapter 5, we investigate the effectiveness of learning by teaching paradigm within the context of crowdsourcing studies, and examine whether teaching a task is more beneficial for crowdworkers than doing the same task for themselves.

Finally, in chapter 6, we describe how the notion of trust may be relevant to teachable agents and whether teaching an agent on a task can make people to delegate similar tasks to the agent where monetary compensation is involved.

7.1 Limitations and Future Work

Performance of the interactive machine learning algorithm proposed in this thesis is based on the cosine similarities obtained from the vector representation of words. We used a compressed variant of Word2Vec trained on a smaller dataset due to performance reasons, which limits the quality of word embeddings used. Future investigations can focus on other word embeddings (eg. GloVe) trained on more relevant and richer dataset for better outcomes. Further, results from chapter 4 shows that effective teaching leads to better machine-learners. However, it remains unclear what characteristics are specific to a good teacher and which factors influence the quality of teaching. It was observed in chapter 5 that letting humans teach can elicit more information from them on a task. However, performance of an interactive machine learner may not depend on the quantity of the information captured alone, as shown in chapter 4. Another important limitation in our study is the lack of animation or personality in the conversational agent leading to a weak test of the engagement mechanism hypothesized to underlie learning outcome effects. Therefore, it will be interesting to explore different modalities of the interaction with teachable agents as opposed to a textual conversational interaction. Follow up studies may involve the use of voice-based agents or embodied agents like physical robots to validate the results in different contexts. Moreover, we do not explore the actual learning outcomes of the teachers after the interaction. An interesting area to explore for follow up work can specifically focus on long and short-term memory changes across longitudinal studies. Finally, more experiments should be conducted to understand the dynamics of trust on teachable agents in the presence of non-teachable agents.

In conclusion, this thesis aims to take one step in the direction to study the learning, perception and trust dynamics of teachable conversational agents. Understanding the breakdowns across these facets will be important for building teachable conversational agents that can reliably learn, be trusted, and benefit human teachers through the conversational interaction.

References

[1] Ag's corpus of news articles. http://www.di.unipi.it/~gulli/AG_corpus_of_news_articles.html. Accessed: 2010-09-30.

[2] Cometomyhead academic news search engine. http://newsengine.di.unipi.it/. Accessed: 2010-09-30.

[3] Ernest Adams. *Fundamentals of game design*. Pearson Education, 2014.

[4] David W Aha, Leonard A Breslow, and Héctor Muñoz-Avila. Conversational case-based reasoning. *Applied Intelligence*, 14(1):9–32, 2001.

[5] Vincent Aleven, Kenneth R Koedinger, and Karen Cross. Tutoring answer explanation fosters learning with understanding understanding. In *Proceedings of the 9th International Conference on Artificial Intelligence in Education*, pages 199–206, 1999.

[6] Saleema Amershi, Maya Cakmak, William Bradley Knox, and Todd Kulesza. Power to the people: The role of humans in interactive machine learning. *AI Magazine*, 35(4):105–120, 2014.

[7] Elisabeth André and Thomas Rist. Presenting through performing: on the use of multiple lifelike characters in knowledge-based presentation systems. *Knowledge-Based Systems*, 14(1-2):3–13, 2001.

[8] Dana Angluin. Queries and concept learning. *Machine learning*, 2(4):319–342, 1988.

[9] Mihael Ankerst, Christian Elsen, Martin Ester, and Hans-Peter Kriegel. Visual classification: an interactive approach to decision tree construction. In *KDD*, volume 99, pages 392–396, 1999.

[10] Christopher G Atkeson, Andrew W Moore, and Stefan Schaal. Locally weighted learning. In *Lazy learning*, pages 11–73. Springer, 1997.

[11] Les E Atlas, David A Cohn, and Richard E Ladner. Training connectionist networks with queries and selective sampling. In *Advances in neural information processing systems*, pages 566–573, 1990.

[12] Frank J Balbach and Thomas Zeugmann. Recent developments in algorithmic teaching. In *International Conference on Language and Automata Theory and Applications*, pages 1–18. Springer, 2009.

[13] John A Bargh and Yaacov Schul. On the cognitive benefits of teaching. *Journal of Educational Psychology*, 72(5):593, 1980.

[14] Amy L Baylor and Yanghee Kim. Simulating instructional roles through pedagogical agents. *International Journal of Artificial Intelligence in Education*, 15(2):95–115, 2005.

[15] Gautam Biswas, Krittaya Leelawong, Daniel Schwartz, Nancy Vye, and The Teachable Agents Group at Vanderbilt. Learning by teaching: A new agent paradigm for educational software. *Applied Artificial Intelligence*, 19(3-4):363–392, 2005.

[16] Susan E Brennan. Lexical entrainment in spontaneous dialog. *Proceedings of ISSD*, 96:41–44, 1996.

[17] Sean Brophy, Gautam Biswas, Thomas Katzlberger, John Bransford, and Daniel Schwartz. Teachable agents: Combining insights from learning theory and computer science. In *Artificial intelligence in education*, volume 50, pages 21–28. Citeseer, 1999.

[18] Maya Cakmak, Crystal Chao, and Andrea L Thomaz. Designing interactions for robot active learners. *IEEE Transactions on Autonomous Mental Development*, 2(2):108–118, 2010.

[19] Maya Cakmak and Manuel Lopes. Algorithmic and human teaching of sequential decision tasks. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[20] Maya Cakmak and Andrea L Thomaz. Optimality of human teachers for robot learners. In *2010 IEEE 9th International Conference on Development and Learning*, pages 64–69. IEEE, 2010.

[21] Maya Cakmak and Andrea L Thomaz. Eliciting good teaching from humans for machine learners. *Artificial Intelligence*, 217:198–215, 2014.

[22] Davide Calvaresi, Yazan Mualla, Amro Najjar, Stéphane Galland, and Michael Schumacher. Explainable multi-agent systems through blockchain technology. In *Proceedings of the 1st International Workshop on eXplanable TRansparent Autonomous Agents and Multi-Agent Systems (EXTRAAMAS 2019)*, 2019.

[23] Giuseppe Carenini and Johanna D Moore. An empirical study of the influence of user tailoring on evaluative argument effectiveness. In *INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE*, volume 17, pages 1307–1314. LAWRENCE ERLBAUM ASSOCIATES LTD, 2001.

[24] Justine Cassell. More than just another pretty face: Embodied conversational interface agents. *Communications of the ACM*, 43(4):70–78, 2000.

[25] 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, 2009.

[26] Li Chen and Pearl Pu. Critiquing-based recommenders: survey and emerging trends. *User Modeling and User-Adapted Interaction*, 22(1-2):125–150, 2012.

[27] Michelene TH Chi, Nicholas De Leeuw, Mei-Hung Chiu, and Christian LaVancher. Eliciting self-explanations improves understanding. *Cognitive science*, 18(3):439–477, 1994.

[28] CIRANO and Pascal Vincent. *Locally weighted full covariance gaussian density estimation*. CIRANO, 2004.

[29] Geraldine Clarebout and Steffi Heidig (née Domagk). *Pedagogical Agents*, pages 2567–2571. Springer US, Boston, MA, 2012.

[30] Herbert H Clark and Deanna Wilkes-Gibbs. Referring as a collaborative process. *Cognition*, 22(1):1–39, 1986.

[31] Leigh Clark, Phillip Doyle, Diego Garaialde, Emer Gilmartin, Stephan Schlögl, Jens Edlund, Matthew Aylett, João Cabral, Cosmin Munteanu, and Benjamin Cowan. The state of speech in hci: Trends, themes and challenges. *arXiv preprint arXiv:1810.06828*, 2018.

[32] Leigh Clark, Abdulmalik Ofemile, Svenja Adolphs, and Tom Rodden. A multimodal approach to assessing user experiences with agent helpers. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 6(4):29, 2016.

[33] Peter A Cohen, James A Kulik, and Chen-Lin C Kulik. Educational outcomes of tutoring: A meta-analysis of findings. *American educational research journal*, 19(2):237–248, 1982.

[34] David Cohn, Les Atlas, and Richard Ladner. Improving generalization with active learning. *Machine learning*, 15(2):201–221, 1994.

[35] Roz Combley. *Cambridge business English dictionary*. Cambridge University Press, 2011.

[36] Benjamin R Cowan, Nadia Pantidi, David Coyle, Kellie Morrissey, Peter Clarke, Sara Al-Shehri, David Earley, and Natasha Bandeira. What can i help you with?: infrequent users' experiences of intelligent personal assistants. In *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services*, page 43. ACM, 2017.

[37] Nils Dahlbäck, QianYing Wang, Clifford Nass, and Jenny Alwin. Similarity is more important than expertise: Accent effects in speech interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 1553–1556. ACM, 2007.

[38] Orlando De Pietro and Giovanni Frontera. Tutorbot: an application aiml-based for web-learning. *Advanced Technology for Learning*, 2(1):29–34, 2005.

[39] Edward L Deci and Richard M Ryan. Self-determination theory. 2012.

[40] Barbara Derriks and Dominique Willems. Negative feedback in information dialogues: identification, classification and problem-solving procedures. *International journal of human-computer studies*, 48(5):577–604, 1998.

[41] Pierre Dillenbourg and John A Self. People power: A human-computer collaborative learning system. In *International Conference on Intelligent Tutoring Systems*, pages 651–660. Springer, 1992.

[42] Kathryn Hershey Dirkin, Punya Mishra, and Ellen Altermatt. All or nothing: Levels of sociability of a pedagogical software agent and its impact on student perceptions and learning. *Journal of Educational Multimedia and Hypermedia*, 14(2):113–127, 2005.

[43] Pedro Domingos and Michael Pazzani. Beyond independence: Conditions for the optimality of the simple bayesian classi er. In *Proc. 13th Intl. Conf. Machine Learning*, pages 105–112, 1996.

[44] D. Christopher Dryer, Chris Eisbach, and Wendy S. Ark. At what cost pervasive? a social computing view of mobile computing systems. *IBM Systems Journal*, 38(4):652–676, 1999.

[45] Thomas Erickson. Designing agents as if people mattered. *Software agents*, pages 79–96, 1997.

[46] Rochelle E Evans and Philip Kortum. The impact of voice characteristics on user response in an interactive voice response system. *Interacting with Computers*, 22(6):606–614, 2010.

[47] Jerry Alan Fails and Dan R Olsen Jr. Interactive machine learning. In *Proceedings of the 8th international conference on Intelligent user interfaces*, pages 39–45. ACM, 2003.

[48] Haiyan Fan and Marshall Scott Poole. What is personalization? perspectives on the design and implementation of personalization in information systems. *Journal of Organizational Computing and Electronic Commerce*, 16(3-4):179–202, 2006.

[49] Donghui Feng, Erin Shaw, Jihie Kim, and Eduard Hovy. An intelligent discussion-bot for answering student queries in threaded discussions. In *Proceedings of the 11th international conference on Intelligent user interfaces*, pages 171–177. ACM, 2006.

[50] Pedro Fialho, Luísa Coheur, Sérgio Curto, Pedro Cláudio, Ângela Costa, Alberto Abad, Hugo Meinedo, and Isabel Trancoso. Meet edgar, a tutoring agent at monserrate. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 61–66, 2013.

[51] Rebecca Fiebrink, Perry R Cook, and Dan Trueman. Human model evaluation in interactive supervised learning. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 147–156. ACM, 2011.

[52] Sally A Goldman and Michael J Kearns. On the complexity of teaching. *Journal of Computer and System Sciences*, 50(1):20–31, 1995.

[53] Arthur C Graesser, Natalie Person, Derek Harter, Tutoring Research Group, et al. Teaching tactics and dialog in autotutor. *International Journal of Artificial Intelligence in Education*, 12(3):257–279, 2001.

[54] Arthur C Graesser, Katja Wiemer-Hastings, Peter Wiemer-Hastings, Roger Kreuz, Tutoring Research Group, et al. Autotutor: A simulation of a human tutor. *Cognitive Systems Research*, 1(1):35–51, 1999.

[55] M Grigoriadou, G Tsaganou, and Th Cavoura. Dialogue-based reflective system for historical text comprehension. In *Workshop on Learner Modelling for Reflection at Artificial Intelligence in Education*, volume 182, 2003.

[56] Andrew Guillory and Jeff A Bilmes. Simultaneous learning and covering with adversarial noise. 2011.

[57] Jaakko Hakulinen, Markku Turunen, Esa-Pekka Salonen, and Kari-Jouko Räihä. Tutor design for speech-based interfaces. In *Proceedings of the 5th conference on Designing interactive systems: processes, practices, methods, and techniques*, pages 155–164. ACM, 2004.

[58] Susumu Harada, Jacob O Wobbrock, Jonathan Malkin, Jeff A Bilmes, and James A Landay. Longitudinal study of people learning to use continuous voice-based cursor control. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 347–356. ACM, 2009.

[59] Alexander G Hauptmann and Paul McAvinney. Gestures with speech for graphic manipulation. *International Journal of Man-Machine Studies*, 38(2):231–249, 1993.

[60] Yugo Hayashi and Koya Ono. Embodied conversational agents as peer collaborators: Effects of multiplicity and modality. In *2013 IEEE RO-MAN*, pages 120–125. IEEE, 2013.

[61] Neil T Heffernan. Web-based evaluations showing both cognitive and motivational benefits of the ms. lindquist tutor. In *Artificial intelligence in education*, pages 115–122, 2003.

[62] Bob Heller and Mike Procter. Conversational agents and learning outcomes: An experimental investigation. In *EdMedia+ Innovate Learning*, pages 945–950. Association for the Advancement of Computing in Education (AACE), 2007.

[63] Kate S Hone and Chris Baber. Designing habitable dialogues for speech-based interaction with computers. *International Journal of Human-Computer Studies*, 54(4):637–662, 2001.

[64] Deanna Hood, Séverin Lemaignan, and Pierre Dillenbourg. When children teach a robot to write: An autonomous teachable humanoid which uses simulated handwriting. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*, pages 83–90. ACM, 2015.

[65] Fatma Howedi and Masnizah Mohd. Text classification for authorship attribution using naive bayes classifier with limited training data. *Computer Engineering and Intelligent Systems*, 5(4):48–56, 2014.

[66] Jiang Hu, Andi Winterboer, Clifford I Nass, Johanna D Moore, and Rebecca Illowsky. Context & usability testing: user-modeled information presentation in easy and difficult driving conditions. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 1343–1346. ACM, 2007.

[67] Shamsi T Iqbal, Eric Horvitz, Yun-Cheng Ju, and Ella Mathews. Hang on a sec!: effects of proactive mediation of phone conversations while driving. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 463–472. ACM, 2011.

[68] Matthias Jerusalem and Ralf Schwarzer. The general self-efficacy scale (gse).[updated 2006 oct 7], 1979.

[69] Jiun-Yin Jian, Ann M Bisantz, and Colin G Drury. Foundations for an empirically determined scale of trust in automated systems. *International Journal of Cognitive Ergonomics*, 4(1):53–71, 2000.

[70] Liangxiao Jiang, Chaoqun Li, and Zhihua Cai. Decision tree with better class probability estimation. *International Journal of Pattern Recognition and Artificial Intelligence*, 23(04):745–763, 2009.

[71] Takayuki Kanda, Takayuki Hirano, Daniel Eaton, and Hiroshi Ishiguro. Interactive robots as social partners and peer tutors for children: A field trial. *Human–Computer Interaction*, 19(1-2):61–84, 2004.

[72] Alice Kerly, Richard Ellis, and Susan Bull. Calmsystem: a conversational agent for learner modelling. In *International Conference on Innovative Techniques and Applications of Artificial Intelligence*, pages 89–102. Springer, 2007.

[73] Ron Kohavi. Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid. In *Kdd*, volume 96, pages 202–207. Citeseer, 1996.

[74] Steven Komarov, Katharina Reinecke, and Krzysztof Z Gajos. Crowdsourcing performance evaluations of user interfaces. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 207–216. ACM, 2013.

[75] Spyros Kousidis, Casey Kennington, Timo Baumann, Hendrik Buschmeier, Stefan Kopp, and David Schlangen. A multimodal in-car dialogue system that tracks the driver's attention. In *Proceedings of the 16th International Conference on Multimodal Interaction*, pages 26–33. ACM, 2014.

[76] Ludovic Le Bigot, Eric Jamet, Jean-François Rouet, and Virginie Amiel. Mode and modal transfer effects on performance and discourse organization with an information retrieval dialogue system in natural language. *Computers in Human Behavior*, 22(3):467–500, 2006.

[77] Ludovic Le Bigot, Patrice Terrier, Virginie Amiel, Gérard Poulain, Eric Jamet, and Jean-François Rouet. Effect of modality on collaboration with a dialogue system. *International Journal of Human-Computer Studies*, 65(12):983–991, 2007.

[78] Kwan Min Lee and Clifford Nass. Designing social presence of social actors in human computer interaction. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 289–296. ACM, 2003.

[79] James C Lester, Sharolyn A Converse, Susan E Kahler, S Todd Barlow, Brian A Stone, Ravinder S Bhogal, et al. The persona effect: affective impact of animated pedagogical agents. In *CHI*, volume 97, pages 359–366. Citeseer, 1997.

[80] James C Lester, Brian A Stone, and Gary D Stelling. Lifelike pedagogical agents for mixed-initiative problem solving in constructivist learning environments. *User modeling and user-adapted interaction*, 9(1-2):1–44, 1999.

[81] David D Lewis and William A Gale. A sequential algorithm for training text classifiers. In *SIGIR'94*, pages 3–12. Springer, 1994.

[82] James R Lewis. Psychometric evaluation of the post-study system usability questionnaire: The pssuq. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 36, pages 1259–1260. SAGE Publications Sage CA: Los Angeles, CA, 1992.

[83] James R Lewis. Ibm computer usability satisfaction questionnaires: psychometric evaluation and instructions for use. *International Journal of Human-Computer Interaction*, 7(1):57–78, 1995.

[84] Clive Loader. *Local regression and likelihood*. Springer Science & Business Media, 2006.

[85] Irene Lopatovska, Katrina Rink, Ian Knight, Kieran Raines, Kevin Cosenza, Harriet Williams, Perachya Sorsche, David Hirsch, Qi Li, and Adrianna Martinez. Talk to me: Exploring user interactions with the amazon alexa. *Journal of Librarianship and Information Science*, page 0961000618759414, 2018.

[86] Giuseppe Lugano. Virtual assistants and self-driving cars. In *2017 15th International Conference on ITS Telecommunications (ITST)*, pages 1–5. IEEE, 2017.

[87] Ewa Luger and Abigail Sellen. Like having a really bad pa: the gulf between user expectation and experience of conversational agents. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 5286–5297. ACM, 2016.

[88] James N MacGregor. The effects of order on learning classifications by example: heuristics for finding the optimal order. *Artificial Intelligence*, 34(3):361–370, 1988.

[89] Dominic W Massaro, Michael M Cohen, Sharon Daniel, and Ronald A Cole. Developing and evaluating conversational agents. In *Human performance and ergonomics*, pages 173–194. Elsevier, 1999.

[90] H David Mathias. A model of interactive teaching. *journal of computer and system sciences*, 54(3):487–501, 1997.

[91] Andrew McCallum, Kamal Nigam, et al. A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, volume 752, pages 41–48. Citeseer, 1998.

[92] Miroslav Melichar and Pavel Cenek. From vocal to multimodal dialogue management. In *Proceedings of the 8th international Conference on Multimodal interfaces*, pages 59–67. ACM, 2006.

[93] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[94] Kathleen K Molnar and Marilyn G Kletke. The impacts on user performance and satisfaction of a voice-based front-end interface for a standard software tool. *International Journal of Human-Computer Studies*, 45(3):287–303, 1996.

[95] Roxana Moreno, Richard E Mayer, Hiller A Spires, and James C Lester. The case for social agency in computer-based teaching: Do students learn more deeply when they interact with animated pedagogical agents? *Cognition and instruction*, 19(2):177–213, 2001.

[96] Maria Moundridou and Maria Virvou. Evaluating the persona effect of an interface agent in a tutoring system. *Journal of computer assisted learning*, 18(3):253–261, 2002.

[97] Krista R Muis, Cynthia Psaradellis, Marianne Chevrier, Ivana Di Leo, and Susanne P Lajoie. Learning by preparing to teach: Fostering self-regulatory processes and achievement during complex mathematics problem solving. *Journal of Educational Psychology*, 108(4):474, 2016.

[98] AC Murray, Dylan M Jones, and CR Frankish. Dialogue design in speech-mediated data-entry: the role of syntactic constraints and feedback. *International Journal of Human-Computer Studies*, 45(3):263–286, 1996.

[99] Clifford Nass and Kwan Min Lee. Does computer-generated speech manifest personality? an experimental test of similarity-attraction. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 329–336. ACM, 2000.

[100] Clifford Nass, Jonathan Steuer, and Ellen R Tauber. Computers are social actors. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 72–78. ACM, 1994.

[101] John F Nestojko, Dung C Bui, Nate Kornell, and Elizabeth Ligon Bjork. Expecting to teach enhances learning and organization of knowledge in free recall of text passages. *Memory & Cognition*, 42(7):1038–1048, 2014.

[102] F Olsson. Bootstrapping named entity recognition by means of active machine learning. *University of Gothenburg*, 2008.

[103] Fredrik Olsson. A literature survey of active machine learning in the context of natural language processing. 2009.

[104] Sharon Oviatt, Colin Swindells, and Alex Arthur. Implicit user-adaptive system engagement in speech and pen interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 969–978. ACM, 2008.

[105] Aannemarie Sullivan Palinscar and Ann L Brown. Reciprocal teaching of comprehension-fostering and comprehension-monitoring activities. *Cognition and instruction*, 1(2):117–175, 1984.

[106] Neil Patel, Sheetal Agarwal, Nitendra Rajput, Amit Nanavati, Paresh Dave, and Tapan S Parikh. A comparative study of speech and dialed input voice interfaces in rural india. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 51–54. ACM, 2009.

[107] Anne Marie Piper and James D Hollan. Supporting medical conversations between deaf and hearing individuals with tabletop displays. In *Proceedings of the 2008 ACM conference on Computer supported cooperative work*, pages 147–156. ACM, 2008.

[108] Reid Porter, James Theiler, and Don Hush. Interactive machine learning in data exploitation. *Computing in Science & Engineering*, 15(5):12–20, 2013.

[109] Kathleen J Price, Min Lin, Jinjuan Feng, Rich Goldman, Andrew Sears, and Julie A Jacko. Motion does matter: an examination of speech-based text entry on the move. *Universal Access in the Information Society*, 4(3):246–257, 2006.

[110] Jason D Rennie, Lawrence Shih, Jaime Teevan, and David R Karger. Tackling the poor assumptions of naive bayes text classifiers. In *Proceedings of the 20th international conference on machine learning (ICML-03)*, pages 616–623, 2003.

[111] Jong-Eun Roselyn Lee, Clifford Nass, Scott Brenner Brave, Yasunori Morishima, Hiroshi Nakajima, and Ryota Yamada. The case for caring colearners: The effects of a computer-mediated colearner agent on trust and learning. *Journal of Communication*, 57(2):183–204, 2006.

[112] N Roy and A McCallum. Toward optimal active learning through sampling estimation of error reduction. int. conf. on machine learning, 2001.

[113] Daisuke Sato, Shaojian Zhu, Masatomo Kobayashi, Hironobu Takagi, and Chieko Asakawa. Sasayaki: augmented voice web browsing experience. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 2769–2778. ACM, 2011.

[114] Stefan Schaffer, Robert Schleicher, and Sebastian Möller. Modeling input modality choice in mobile graphical and speech interfaces. *International Journal of Human-Computer Studies*, 75:21–34, 2015.

[115] Alex Sciuto, Arnita Saini, Jodi Forlizzi, and Jason I Hong. Hey alexa, what's up?: A mixed-methods studies of in-home conversational agent usage. In *Proceedings of the 2018 Designing Interactive Systems Conference*, pages 857–868. ACM, 2018.

[116] Burr Settles. Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2009.

[117] Burr Settles, Mark Craven, and Soumya Ray. Multiple-instance active learning. In *Advances in neural information processing systems*, pages 1289–1296, 2008.

[118] H Sebastian Seung, Manfred Opper, and Haim Sompolinsky. Query by committee. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 287–294. ACM, 1992.

[119] E Shaw, WL Johnson, and R Ganeshan. Pedagogical agents on the web. in international conference on autonomous agents, 1999.

[120] Patrice Y Simard, Saleema Amershi, David M Chickering, Alicia Edelman Pelton, Soroush Ghorashi, Christopher Meek, Gonzalo Ramos, Jina Suh, Johan Verwey, Mo Wang, et al. Machine teaching: A new paradigm for building machine learning systems. *arXiv preprint arXiv:1707.06742*, 2017.

[121] Venkatesh Sivaraman, Dongwook Yoon, and Piotr Mitros. Simplified audio production in asynchronous voice-based discussions. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 1045–1054. ACM, 2016.

[122] Kate Taylor and Simon Moore. Adding question answering to an e-tutor for programming languages. In *International Conference on Innovative Techniques and Applications of Artificial Intelligence*, pages 193–206. Springer, 2006.

[123] Cynthia A Thompson, Mehmet H Goker, and Pat Langley. A personalized system for conversational recommendations. *Journal of Artificial Intelligence Research*, 21:393–428, 2004.

[124] Simon Tong and Daphne Koller. Active learning for structure in bayesian networks. In *International joint conference on artificial intelligence*, volume 17, pages 863–869. Citeseer, 2001.

[125] Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *Journal of machine learning research*, 2(Nov):45–66, 2001.

[126] David Traum, Priti Aggarwal, Ron Artstein, Susan Foutz, Jillian Gerten, Athanasios Katsamanis, Anton Leuski, Dan Noren, and William Swartout. Ada and grace: Direct interaction with museum visitors. In *International conference on intelligent virtual agents*, pages 245–251. Springer, 2012.

[127] Aditya Vashistha, Pooja Sethi, and Richard Anderson. Respeak: A voice-based, crowd-powered speech transcription system. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 1855–1866. ACM, 2017.

[128] Danli Wang, Jie Li, Jie Zhang, and Guozhong Dai. A pen and speech-based storytelling system for chinese children. *Computers in Human Behavior*, 24(6):2507–2519, 2008.

[129] Malcolm Ware, Eibe Frank, Geoffrey Holmes, Mark Hall, and Ian H Witten. Interactive machine learning: letting users build classifiers. *International Journal of Human-Computer Studies*, 55(3):281–292, 2001.

[130] Noreen M Webb. Predicting learning from student interaction: Defining the interaction variables. *Educational psychologist*, 18(1):33–41, 1983.

[131] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657, 2015.