How AI Wins Friends and Influences People in Repeated Games with Cheap Talk

Mayada Oudah
Dept. of Computer Science
Masdar Institute
Abu Dhabi, UAE
oudah.mayada@gmail.com

Talal Rahwan
Dept. of Computer Science
Khalifa Univ. of Sci. and Tech.
Abu Dhabi, UAE
trahwan@masdar.ac.ae

Tawna Crandall
Computer Science Dept.
Brigham Young University
Provo, UT, USA
tawnac@gmail.com

Jacob W. Crandall
Computer Science Dept.
Brigham Young University
Provo, UT, USA
crandall@cs.byu.edu

Abstract

Research has shown that a person's financial success is more dependent on the ability to deal with people than on professional knowledge. Sage advice, such as "if you can't say something nice, don't say anything at all" and principles articulated in Carnegie's classic *How to Win Friends and Influence People*, offer trusted rules-of-thumb for how people can successfully deal with each other. However, alternative philosophies for dealing with people have also emerged. The success of an AI system is likewise contingent on its ability to win friends and influence people. In this paper, we study how AI systems should be designed to win friends and influence people in repeated games with cheap talk (RGCTs). We create several algorithms for playing RGCTs by combining existing behavioral strategies (what the AI does) with signaling strategies (what the AI says) derived from several competing philosophies. Via user study, we evaluate these algorithms in four RGCTs. Our results suggest sufficient properties for AIs to win friends and influence people in RGCTs.

Introduction

In his classic book *How to Win Friends and Influence People*, Dale Carnegie argued that a person's financial success is impacted more by the ability to "deal with people" than by professional knowledge (Carnegie 1937, p. 15)[1]. However, so-called people skills are not easy to come by. For many of us, it takes years (and even a lifetime) of guidance and practice to learn the "fine art" of getting along with others, particularly in situations in which other people's interests are not fully aligned with our own.

As AI matures, autonomous agents will perform more tasks in behalf of their human stakeholders. Many of these tasks will require these agents to repeatedly interact with other people (apart from their stakeholders) who may not share all of their preferences. To be successful in such scenarios, autonomous agents must, like humans, be able to win friends and influence people.

In this paper, we study how an AI can develop successful long-term relationships, modeled as repeated games with cheap talk (RGCTs), with people. Dealing successfully with people, we argue, entails two properties. First, a successful AI should obtain high material payoffs for its stakeholder, which requires it to effectively *influence* the behavior of people with whom it interacts. We refer to this property as *influencing people*. Second, a successful AI should *win friends*, meaning that the people with whom it interacts should both think highly of it and desire to continue associating with it. In short, the success of an AI in RGCTs is determined by its ability to both win friends and influence people.

An AI's ability to win friends and influence people in RGCTs depends on both its *behavioral strategy* (what it does) and its *signaling strategy* (what it says). While behavior generation in repeated games has been well studied, effectively signaling in RGCTs is less understood. To begin to address this shortcoming, we derive several algorithms for RGCTs by combining existing behavioral strategies with signaling strategies based on known philosophies for dealing with people, including Thumper's Rule (*if you can't say something nice, don't say anything at all*), Carnegie's Principles (Carnegie 1937), and other alternative theories. Via user studies, we then evaluate the abilities of these algorithms to win friends and influence people across four RGCTs.

This paper has two primary contributions. First, we propose that, when interacting with people in RGCTs, algorithms should be evaluated with respect to both winning friends and influencing people, rather than the single metric class (payoff maximization) traditionally considered in repeated games. Second, our results suggest sufficient properties for winning friends and influencing people in RGCTs. These results show that an algorithm that (1) quickly learns an effective behavioral strategy while using a signaling strategy built on both (2) Carnegie's Principles and (3) explainable AI (XAI) (Gunning 2016) was more successful at winning friends and influencing people than algorithms that lacked any of those characteristics. This finding has important implications for the design of algorithms that interact

[1]Carnegie's full statement is worth noting: "Dealing with people is probably the biggest problem you face, especially if you are in business. Yes, and that is also true if you are a housewife, architect or engineer. Research done a few years ago under the auspices of the Carnegie Foundation for the Advancement of Teaching uncovered a most important and significant fact–a fact later confirmed by additional studies made at the Carnegie Institute of Technology. These investigations revealed that even in such technical lines as engineering, about 15 percent of one's financial success is due to one's technical knowledge and about 85 percent is due to skill in human engineering–to personality and the ability to lead people."

(a) Prisoner's Dilemma

	X	Y
A	60, 60	0, 100
B	100, 0	20, 20

(b) Chicken

	X	Y
A	0, 0	100, 33
B	33, 100	84, 84

(c) Alternator Game

	X	Y	Z
A	0, 0	35, 70	100, 40
B	70, 35	10, 10	45, 30
C	40, 100	30, 45	40, 40

(d) Endless

	X	Y
A	33, 67	67, 100
B	0, 33	100, 0

Table 1: Payoff matrices of four normal-form games. In each round, Player 1 selects the row, while Player 2 selects the column. The resulting cell of the matrix specifies the payoffs obtained by players 1 and 2, respectively, in the round.

with people who do not share the AI's preferences.

Repeated Games with Cheap Talk

We study repeated interactions between an AI and a person. In behavioral economics, mathematical biology, psychology, sociology, and political science, associations between intelligent entities are commonly modeled with normal-form games. Thus, repeated normal-form games are a natural setting to study long-term relationships between a human and an AI when their preferences are not fully aligned.

A two-player repeated normal-form game, played by players i and $-i$, proceeds as a sequence of rounds. In each round, each player chooses an action from a finite set. Let $A = A_i \times A_{-i}$ be the set of joint actions available, where A_i and A_{-i} are the action sets of players i and $-i$, respectively. When joint action $\mathbf{a} = (a_i, a_{-i}) \in A$ is played, the players receive the finite rewards $r_i(\mathbf{a})$ and $r_{-i}(\mathbf{a})$, respectively. In this paper, we assume perfect information games, wherein the players are aware of the actions and payoffs of both players. We also assume that the number of rounds in the game is unknown to both players.

Examples of normal-form games are shown in Table 1. While each game models a different conflict between the players, each game requires the players to decide whether to try to cooperate with their partner, exploit their partner, or defend themselves against being exploited.

Though repeated normal-form games provide a natural setting for studying human-AI partnerships, they do not facilitate an important aspect of many human relationships– the ability to communicate using *cheap talk*, which is a costless, non-binding, and unverifiable form of communication. Cheap talk has been shown to facilitate cooperation in repeated games played by human players (Charness and Grosskopf 2004; Crawford 1998; Crawford and Sobel 1982; Farrell 1987; Farrell and Rabin 1996; Green and Stokey 2007). In this paper, we consider how to create autonomous agents that can use such communication to cooperate with people in *repeated games with cheap talk* (RGCTs).

In each round of an RGCT, each player sends a message to its partner before acting. That is, at the beginning of round t, player i sends message $m_i(t)$ to player $-i$, who simultaneously sends messages $m_{-i}(t)$ to i. Only after sending $m_i(t)$ can i view $m_{-i}(t)$ (and vice versa). The players then select actions for the round as in conventional repeated games.

Thus, a strategy in an RGCT is a combination of a signaling and a behavioral strategy. Let M_i be the (possibly infinite) set of messages available to player i. Then, let ϕ_i^t be a probability distribution over M_i denoting player i's *signaling policy* in round t, and let π_i^t be a probability distribution over A_i denoting player i's *behavioral policy* in round t. Then, the tuple (ϕ_i^t, π_i^t) is player i's policy in round t. Since players should likely respond to past messages and actions used by their partner, player i's *policy* (ϕ_i^t, π_i^t) in round t is likely contingent on some or all of the history of the game, which is defined by the messages and actions taken by both players in all previous rounds. Thus, player i's *strategy* is defined by the policy it would use in all possible game states, where game states are defined by the full history of the game.

Evaluating Algorithms in RGCTs

A successful algorithm should maximize the utilities of players that use it. However, in RGCTs played with people, it is sometimes unclear what to maximize. In such scenarios, we argue that algorithms should be evaluated in terms of two sets of metrics: *influencing people* and *winning friends*.

Influencing People

Metrics for *influencing people* measure an algorithm's ability to influence its partner's behavior so that it achieves high rewards. One direct metric of influence, which we call *Partner Cooperation*, is the proportion of rounds that the algorithm's partner *cooperates* with it. We say that player $-i$ cooperates with player i in round t if $a_{-i}^t \in \arg\max_{b \in A_{-i}} r_i(a_i^t, b)$, where a_{-i}^t is the action taken by player $-i$ in round t. In words, player $-i$ cooperated in round t if its action maximized the reward received by player i in round t given the action played by i.

Since influence typically leads to high payoffs, the total reward, called *material payoff*, achieved by a player throughout a repeated game, is an alternative, but less direct, metric of influence. Player i's material payoff in an RGCT with T rounds is its average per-round payoff U_i. Let r_i^t be player i's reward in round t. Then, $U_i = \frac{1}{T} \sum_{t=1}^{T} r_i^t$.

Traditionally, success in repeated games has been defined by the ability to maximize payoffs. However, since achieving, defining, and measuring optimal behavior in repeated games is difficult (Axelrod 1984; de Farias and Megiddo 2004; Crandall 2014), much work has focused on developing algorithms that meet certain criteria, such as convergence to Nash equilibria (Fudenberg and Levine 1998; Hu and Wellman 1998; Littman 2001; Bowling and Veloso 2002) or Pareto optimal solutions (Powers and Shoham 2005), minimizing regret (Foster and Vohra 1999; Bowling 2004; Greenwald and Jafari 2003; Fudenberg and Levine 1998), and being secure (Fudenberg and Levine 1998; Powers and Shoham 2005). Despite the appeal of these metrics, we do not consider them in this paper since they often do not correlate with high material payoffs (de Farias and Megiddo 2004; Arora, Dekel, and Tewari 2012; Crandall 2014).

Winning Friends

Metrics of *winning friends* measure social consequences not necessarily reflected in a single repeated game. Many AIs repeatedly interact with many different people. People's perceptions of the AI determine whether they encourage others to enter into relationships with the AI. Furthermore, in practice, people often can choose whether or not they continue associating with the AI. As such, human perceptions of the AI could be as important (or even more so) than the actual payoffs obtained by the AI in any given RGCT.

We measure an AI's ability to win friends in two ways. First, we measure how much people want to continue associating with it using the *Attraction Index*, a metric derived from responses of human participants in user studies. After participants play an RGCT, we ask them if they would like to interact with their partner again. $Agn(j) = 1$ if the participant answered *yes* after associating with player j, and $Agn(j) = 0$ otherwise. Additionally, after a participant plays four RGCTs (each with a different partner), we ask them which of their partners was their favorite. $Fav(j) = 1$ if the participant chose player j, and $Fav(j) = 0$ otherwise. Then, the *Attraction Index* of player j as assessed by the participant is $Agn(j) + Fav(j)$. Higher average values over all participants indicate a greater ability to maintain friends.

Second, we measure the character reputation the AI forges with human partners. To do this, we ask participants to rate their partners (on a 5-point Likert scale) with respect to eight attributes: likable, intelligent, cooperative, trustworthy, forgiving, selfish, vengeful, and tendency to bully. To summarize the AI's ability to create a positive character reputation, we average all eight ratings, inverting the last three negative attributes. We call this metric the *Character Index*.

Algorithms for RGCTs

Many algorithms have been proposed and analyzed for repeated games (Bouzy and Metivier 2010; Hoen et al. 2006; Shoham, Powers, and Grenager 2007; Hernandez-Leal et al. 2017). RGCTs have been less studied. Most work in RGCTs has been limited to human-human interactions (Charness and Grosskopf 2004; Crawford 1998; Crawford and Sobel 1982; Farrell 1987; Farrell and Rabin 1996; Green and Stokey 2007). However, a new algorithm, called S#, was recently shown to match human cooperation in several RGCTs (Oudah et al. 2015; Crandall et al. 2017). While this prior work demonstrated that a particular signaling and behavioral strategy could induce people to cooperate with an AI, it did not thoroughly study what makes the algorithm successful. Thus, in this paper, we study how various behavioral and signaling strategies jointly impact an AI's ability to both win friends and influence people by comparing the performance of a variety of algorithms via user studies. These algorithms are formed by combining together two existing behavioral strategies with various signaling strategies.

Selected Behavioral Strategies

From the many algorithms that have been created for repeated games, we selected S++ (Crandall 2014) and EEE (de Farias and Megiddo 2004) to generate behavioral strategies due to their distinct behavior and performance attributes. Both algorithms are expert algorithms that pre-compute a set of expert strategies from the game's payoff matrix, and then learn over time which expert strategy to follow. S++ uses an enhanced version of aspiration-learning (Karandikar et al. 1998) to choose among these experts. We selected this algorithm because it was the highest performing algorithm in a recent comparison of 25 algorithms in repeated games (Crandall et al. 2017). It often quickly learns to reciprocate defection and cooperation, and to convey a fair and demanding expectation to its partner. Our implementation of S++ was identical to the implementation used by Crandall et al. (2017).

On the other hand, EEE uses an ε-greedy mechanism for selecting which expert to follow in each round. In the same comparison of 25 algorithms for repeated games, it had a lower, but still adequate, level of performance than S++. EEE is more lenient toward its partner than S++, as (particularly during early rounds of a game) it can be convinced to follow experts that produce higher payoffs to its partner than to itself. As such, its partners tend to receive higher payoffs than S++'s. Details of our implementation of EEE are given in the supplementary material.

While these two algorithms differ with respect to both behavior and performance, both algorithms produce coherent strategies within a relatively small number of rounds of interaction. This makes these algorithms potentially acceptable for interacting with people.

Table 2: Algorithmic events and corresponding speech categories. Table 4 maps these speech categories to speech acts.

Algorithmic Events	**Speech Category**
Select a new behavioral strategy	0-4
Accept the partner's proposal	5
Reject the partner's proposal (due to distrust)	6
Reject the partner's proposal (it seems unfair)	7
Belief that both players can get higher payoffs	8
The partner defected	9
The partner profited from its defection	10
The alg. punished its guilty partner	11
The alg. forgives its partner	12
Last round's payoff was satisfactory to the alg.	13
The game begins; the alg. is initialized.	14

Adding Signaling Strategies

S++ and EEE are both designed for repeated games. They are not equipped for RGCTs, as they do not produce or respond to cheap talk. However, recent work (Oudah et al. 2015; Crandall et al. 2017) provides one mechanism for generating and responding to speech acts using existing behavioral strategies. In that work, S++'s internal state is used to identify game-invariant *algorithmic events* (Table 2) related to proficiency assessment, fairness assessment, behavioral expectations, and social mechanisms such as punishment and forgiveness. This algorithm is called S#. In the same way, EEE can also be used to identify the same game-

Table 3: A subset of Carnegie's Principles (Carnegie 1937), grouped and reworded for brevity.

ID	Carnegie's Principles
A	Don't criticize, condemn, or complain.
B	If you must, call attention to other people's mistakes indirectly. Make a fault seem easy to correct.
C	Give sincere appreciation. Praise improvements.
D	Talk in terms of the other person's interest.
E	Be sympathetic with the other person's ideas and desires.
F	If you're wrong, admit it quickly and emphatically.
G	Begin in a friendly way.
H	Ask questions instead of giving direct orders.
I	Let the other person feel the idea is his or hers.
J	Give the other person a fine reputation to live up to.

invariant *algorithmic events* from which speech acts can be generated and from which the proposal of one's partner can be used to select actions (see the supplementary material). We refer to this new algorithm as EEE#. S# and EEE# differ from S++ and EEE only with respect to their ability to generate and respond to speech acts.

By mapping game-invariant algorithmic events to speech categories (Table 2), behavioral strategies identify cheap talk that is consistent with the algorithm's internal state. To complete the signaling strategy, we need only specify speech acts for each speech category. We create distinct signaling strategies by varying the speech acts in each speech category.

We consider four different signaling strategies, which we refer to as *personas*. Rather than basing signaling strategies on emotion (Breazeal and Scassellati 1999) or personality taxonomies (von der Putten, Kramer, and Gratch 2010), we derives these personas from four popularized rules-of-thumb defining how successful people should treat each other. The first of these personas is derived from the principles presented in Dale Carnegie's classic *How to Win Friends and Influence People* (Carnegie 1937). These principles are summarized in Table 3. We call this persona CARNEGIE. CARNEGIE seeks to avoid criticizing, complaining, or condemning its partners, while respectfully building them up. Table 4 lists an example speech act for each speech category used by CARNEGIE. The table also indicates how each speech act relates to Carnegie's Principles in Table 3.

While Carnegie's Principles have been widely accepted as winning principles for dealing with people, a counter-culture is prevalent in society. For example, it has become somewhat commonplace for politicians, many of whom would be considered successful by many standards, to criticize and belittle their political opponents and associates. This counter-culture eschews political correctness in favor of bluntness (perhaps because there is no time for such niceties), seeks to pull others down rather than build them up, and promotes one's own self. In short, this counter-culture espouses principles opposite to Carnegie's Principles.

To learn how adopting this philosophy impacts an AI's ability to win friends and influence people, we created a second signaling strategy that seeks to emulate it. We call this persona BIFF[2] after the fictional character Biff Tannen in *Back to the Future*. BIFF belittles its partner, blames its partner for undesirable outcomes, takes credit for good outcomes, and talks in terms of its own interests. Example speech acts for BIFF are also given in Table 4.

Our third persona, which also contrasts BIFF, adheres to Thumper's Rule as expressed in the Disney film *Bambi*: "If you can't say something nice, don't say anything at all." While BIFF says things that are not nice, this third persona, called THUMPER, refrains from saying anything at all. Thus, algorithms that use this persona listen to their partner, but are nonverbal. They do not generate speech acts themselves.

Finally, while CARNEGIE and BIFF both express emotions and opinions through speech acts, our fourth persona does not. This persona, named SPOCK after the fictional *Star Trek* character, encodes a stereotypical robot that expresses facts, but not emotions and opinions. Though SPOCK does not express appreciation or build others up, it adheres to several of Carnegie's Principles (Table 3), particularly with regards to not criticizing, condemning, or complaining.

We combined the two selected behavioral strategies with each of the four personas to form eight distinct algorithms, which we refer to as S#-CARNEGIE, S#-BIFF, S#-THUMPER, S#-SPOCK, EEE#-CARNEGIE, EEE#-BIFF, EEE#-THUMPER, and EEE#-SPOCK. In the next section, we describe a user study designed to evaluate how well these algorithms win friends and influence people.

User Study 1

In this user study, participants played RGCTs with the eight algorithms described in the previous section. We describe the experimental design of the study, followed by the results.

Experimental Design

The user study was a 2×4 mixed factorial design in which behavioral strategy (S# and EEE#) was a between-subjects variable and persona (CARNEGIE, BIFF, SPOCK, and THUMPER) was a within-subjects variable.

Experimental Protocol Ninety-six people (average age: 26.7 years) at Masdar Institute (Abu Dhabi, UAE) volunteered to participate in this study. Each participant was randomly assigned to play RGCTs with either S# or EEE#, such that 48 subjects were assigned to each condition. Each participant played the four RGCTs shown in Table 1 in the order shown in the table. In each game, the participant was paired with a different persona, though they were not told if they were paired with another person or an AI. The order the participants were exposed to the personas was fully counterbalanced across participants to nullify ordering effects.

The games were played through a GUI on a desktop computer. Participants were first trained on how to play the game through the GUI, of which a full description is provided in the supplementary material. At the start of each round, the participant created and sent a chat message to the other

[2]We use the names of fictional characters from popular films to help the reader remember the signaling strategies.

Table 4: Example speech acts for the signaling strategies CARNEGIE, SPOCK, and BIFF for each speech category (Table 2). The full set of speech acts for each category is given in the supplementary material. CP denotes the Carnegie Principles (partially) invoked by a speech act (Table 3). ¬ denotes that the speech act directly contradicts a principle.

Cat.	Example speech acts for CARNEGIE	Example speech acts for SPOCK	Example speech acts for BIFF
0	Let's always play <solution>.	Let's always play <solution>.	Let's always play <solution>.
1	Let's alternate between <solution> and <solution>.	Let's alternate between <solution> and <solution>.	Let's alternate between <solution> and <solution>.
2	This round, let's play <solution>.	This round, let's play <solution>.	This round, let's play <solution>.
3	if we can agree, we'll both benefit. (CP: D)	u will get punished if u don't follow this plan. (CP: D)	listen to me or U WILL REGRET BEING BORN. (CP: ¬A, ¬H)
4	let's explore other options that may be better for us. (CP: A, D)	I am going to explore other options. (CP: A)	... sigh, u aren't letting me get as many points as I deserve. (CP: ¬A, ¬D)
5	good idea. as expected from a generous person like u. I accept your proposal. (CP: I, J)	I accept your proposal.	even u managed to see the obvious. I accept your proposal. (CP: ¬I)
6	good proposal. if u show that u are trustworthy, I will consider accepting it in the future. (CP: B, D, J)	I don't accept your proposal. (CP: A)	u r SLEAZY. Can't trust u. (CP: ¬A, ¬B, ¬E)
7	a fairer proposal would work to your benefit. (CP: A, B, D)	I don't accept your proposal. (CP: A)	as for your proposal: r u kidding me? it is very unfair! VERY unfair!! (CP: ¬A, ¬B, ¬E)
8	your payoffs can be higher than this. (CP: A, B, D)	we can get higher payoffs than this. (CP: A, B, D)	I need u to listen to me. (CP: ¬D)
9	what u did is totally understandable, though it will not benefit u in the long run. (CP: D, E)	that was not what I expected. (CP: A, B)	selfish traitor! you've treated me very unfairly. (CP: ¬A, ¬D, ¬J)
10	in the next round comes the expected penalty, but we can then return to cooperating. (CP: A)	I will punish u for this. (CP: D)	u will regret having backstabbed me. (CP: ¬A, ¬E)
11	I'm really sorry I had to do that. (CP: F)	I punished u. (CP: D)	THAT was exactly what the likes of u deserve. (CP: ¬C)
12	let's move on. I am sure we can get along. (CP: A, B)	I am done punishing u. (CP: D)	u have been unimaginably selfish, but I will look past it for now. (CP: ¬A, ¬B, ¬E, ¬I)
13	excellent! Thanks for cooperating with me. (CP: C)		I make great deals. (CP: ¬I)
14	Hey there! What do you think would be a fair outcome? (CP: G, H)		Hello, I would like to make lots of money in this game. (CP: ¬D)

player (the computer algorithm). Participants could say anything they wanted, except that they were not allowed to reveal or try to determine the identify of their partner through these messages. After sending the message, the chat message sent by the participant's partner was displayed on the GUI and spoken to the participant over headsets using a computerized voice. The participant then selected an action and viewed the results of the round of play. This process continued for 50 rounds, though neither player was told how many rounds the game would last to avoid end-game effects. After each game, participants completed a survey, which asked questions related to the Attraction and Character Indices described previously.

Participants were told that they would be paid proportionally to the rewards they received in the games they played. Overall, participants typically received between $15-25 depending on performance. The amount of money earned by participants was displayed on the GUI. Participants were not told the identity of their partners. To conceal whether they were partnered with human or computer players, participants were recruited in groups of four. Computers were arranged so that the participants were not visible to each other.

Metrics Table 5 summarizes the metrics used to evaluate the algorithms' abilities to win friends and influence people. To compare the relative performance of algorithms across games, we use the standardized z-score for each metric. For example, a player's relative material payoff for repeated game g is given by $\frac{U_i - U(g)}{\sigma(g)}$, where $U(g)$ and $\sigma(g)$ are the mean and standard deviations of material payoffs achieved by players in game g.

Table 5: Performance metrics used in the study.

Influencing People	Winning Friends
1. Partner Cooperation	1. Attraction Index
2. Material Payoffs	2. Character Index

Games Our goal is to identify algorithms that win friends and influence people in general, and not just in certain games. While we are limited to evaluating algorithms in a handful of scenarios, we carefully selected games to generalize distinct types of conflicts between players. To do this, we selected games using the periodic table of games (Robinson and Goforth 2005), which classifies normal-form games into six payoff families. The four RGCTs included in our study (Table 1) were drawn from distinct payoff families that encoded the most challenging conflicts. By and large, the results were consistent across games.

Results

Relative comparisons of the algorithms with respect to the four individual metrics are shown in Figure 1. Figure 2 summarizes the results of the study by showing the relative performance of the eight algorithms with respect to both win-

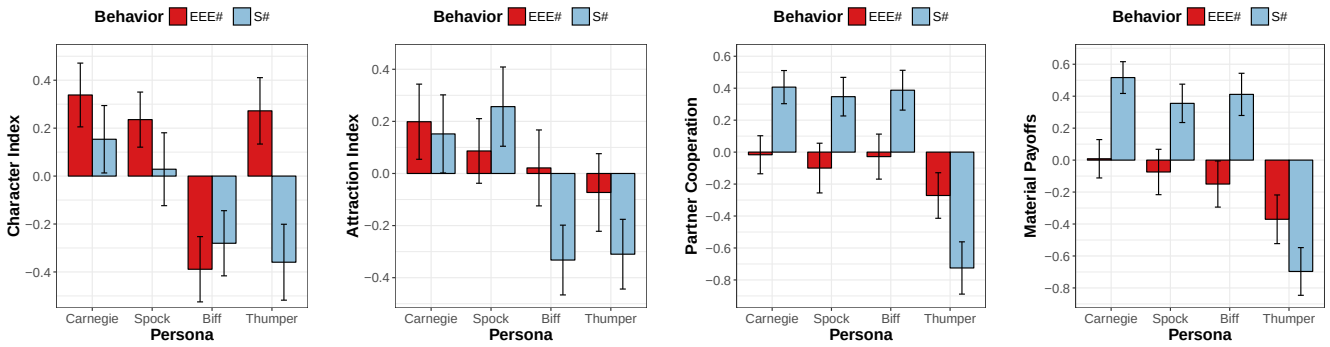

Figure 1: Measures of winning friends (*Character Index* and *Attraction Index*) and influencing people (*Partner Cooperation* and *Material Payoffs*) in the first user study. Results are displayed as standardized z-scores to illustrate relative performance, with error bars giving the standard error of the mean. The unit of each axis is the standard deviation from the mean.

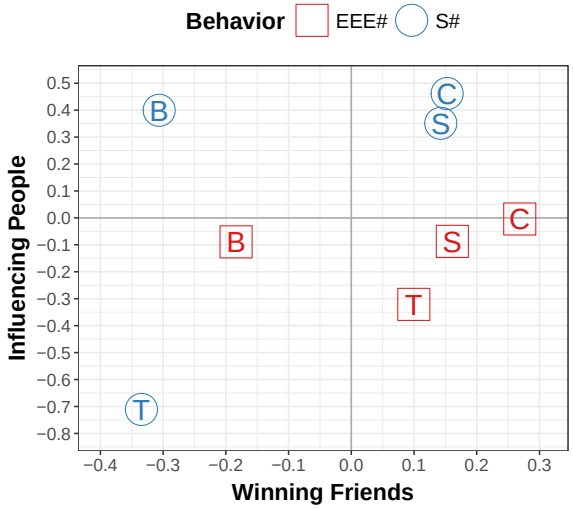

Figure 2: A summary of results of the first user study. *Influencing People* (y-axis) is the average of Material Payoffs and Partner Cooperation, while *Winning Friends* (x-axis) is the average of the Character and Attraction Indices. Axes units are standard deviations from the mean. Signaling strategies (personas) are represented by their first letters.

ning friends and influencing people. In the interest of space, we focus on a handful of results, each of which is supported by a full statistical analysis, using the Aligned Rank Transform (Wobbrock et al. 2011) for analyzing non-parametric factorial data with repeated measures, provided in the supplementary material. We also reflect on the importance of Carnegie's Principles and Thumper's Rule.

Primary Outcomes An algorithm's ability to successfully influence people was driven by both its behavioral and signaling strategies. We note two outcomes related to influence. First, algorithms that generated cheap talk had higher influence than those that did not. Across both behaviors, THUMPER had less influence with respect to both material payoffs and partner cooperation than the other three personas ($p < 0.001$). Second, given a verbal signaling strategy, S# outperformed EEE#. For example, with respect to material payoffs, S# outperformed EEE# given the personas CARNEGIE, BIFF, and SPOCK ($p < 0.001$, $p = 0.001$, and $p = 0.015$, respectively). Results for partner cooperation were similar, though the difference between S#-BIFF and EEE#-BIFF was only marginally significant ($p = 060$).

Further analysis of the results indicates why S# outperformed EEE# given a verbal persona: EEE# is often content with solutions that give its partner a much higher payoff than it receives itself, whereas S# is not. Across all games played, EEE# reciprocated defection immediately after being exploited in a round just 27% of the time, while S# reciprocated defection after being exploited 76% of the time. As a result, human players were forced to cooperate with S# to receive high payoffs, while they were often able to get away with exploiting EEE#. This translated into higher payoffs for participants when they associate with EEE# than with S# ($p < 0.001$), a result that held regardless of the signaling strategy. On the other hand, S# received higher payoffs when paired with people than did EEE#.

Even though people earned more money when paired with EEE# than S#, EEE# was not universally better than S# with respect to winning friends. EEE# had a marginally statistically higher Character Index ($p = 0.052$) over all personas, but there was no statistically significant difference with respect to the Attraction Index ($p = 0.232$). Main interaction effects between behavior and signaling strategy showed that the ability to win friends was impacted by the joint signaling and behavioral strategies. While S#-BIFF and S#-THUMPER performed poorly with respect to both the Character Index and the Attraction Index, there were no statistically significant differences between S# and EEE# given the personas CARNEGIE and SPOCK. Though participants received lower rewards when partnered with S#-CARNEGIE and S#-SPOCK, they still rated these algorithms as highly as EEE#-CARNEGIE and EEE#-SPOCK with respect to the Character and Attraction Indices.

Critique of Carnegie's Principles Figures 1–2 demonstrate the usefulness of Carnegie's Principles when implementing signaling strategies. Recall that both CARNEGIE and SPOCK adhere to some of Carnegie's Principles. While CARNEGIE embraces these principles to a large degree, SPOCK conforms only to a subset of these principles, in particular with respect to not complaining, criticizing, or condemning others. Across all four metrics, these two signaling strategies performed very well compared to the other signaling strategies. However, there was essentially no distinction between CARNEGIE and SPOCK with respect to any metric. These results suggest that not going directly against Carnegie's Principles is important, though some of these principles may be more important than others.

Critique of Thumper's Rule Common convention suggests that "if you can't say something nice, don't say anything at all." A comparison between the BIFF and THUMPER signaling strategies suggests that this advice is not universally true, and is even, with respect to some metrics, misguided. In our study, THUMPER was outperformed by BIFF with respect to both metrics of influence ($p < 0.001$). The results are less conclusive with respect to winning friends. EEE#-THUMPER did outperform EEE#-BIFF with respect to the Character Index (interestingly, users felt, in particular, that EEE#-BIFF was not very intelligent). However, in all other comparisons related to the Character and Attraction Indices, THUMPER did not outperform BIFF.

Together, these results suggest that, if one must choose between silence and communicating albeit rudely, erring on the side of communicating is likely more beneficial with respect to influence in RGCTs. However, rude communication may lower one's character reputation, and hence may not be beneficial with respect to winning friends.

Summary of Results for User Study 1

Across all four metrics, only S#-CARNEGIE and S#-SPOCK were not statistically outperformed with respect to any measure. This suggests that these two algorithms provide a nice balance of winning friends and influencing people. Coupling a behavioral strategy that learns quickly and effectively with a signaling strategy built on Carnegie's Principles (or at least not violating them) appears to result in a strategy that wins friends and influences people.

However, these results raise further questions. In the next section, we seek to better understand what makes a signaling strategy successful. In particular, we investigate the importance of explainable AI.

User Study 2

In the previous study, S# and EEE# were endowed with *explainable AI* (XAI) (van Lent, Fisher, and Mancuso 2004; Gunning 2016), which allowed them to express strategies at levels people understood, and to comprehend their partners' proposals. In normal-form games, communicating strategies is relatively simple, as actions can be described by simply naming the rows or columns in the payoff matrix. However, XAI is not so easily achieved in more complex domains in which humans communicate at a high level of abstraction.

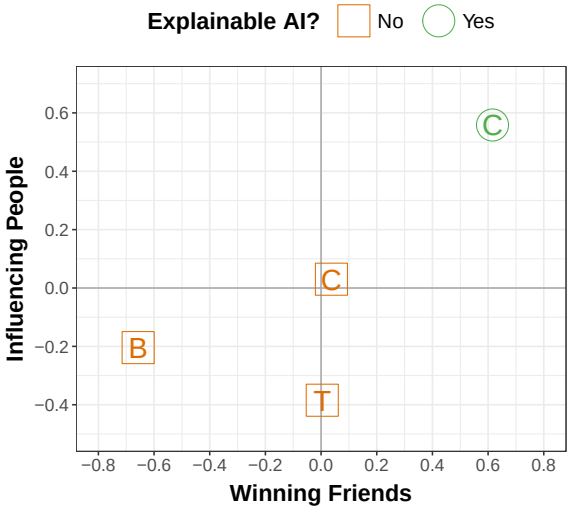

Figure 3: An overview of the results of the second user study. See Figure 2 for axes descriptions.

We conducted a second user study to understand the importance of XAI in signaling strategies. In this study, we compared the performance of S#-CARNEGIE to algorithms not equipped with XAI (NXAI), including S#-CARNEGIENXAI, S#-BIFFNXAI, and S#-THUMPERNXAI. These algorithms were equivalent to similarly named algorithms used in the first study, except that they could not understand their partner's proposals, nor could they voice speech acts that communicated high-level plans. The speech acts used by S#-CARNEGIENXAI and S#-BIFFNXAI are given in the supplementary material.

Forty-eight people at Brigham Young University (Provo, UT, USA) volunteered to participate in this second study. We used the same experimental protocol in this study as in the first study. Each participant interacted with each of the four algorithms in the same four RGCTs (Table 1).

Results are summarized in Figure 3. S#-CARNEGIE outperformed the other algorithms with respect to all four metrics. On the other hand, there was no statistical separation between S#-CARNEGIENXAI and S#-THUMPERNXAI with respect to any of the metrics. As such, it appears that XAI accounted for much of S#-CARNEGIE's ability to win friends and influence people in the first user study. We note, however, that even without XAI, not violating Carnegie's Principles was still important with respect to winning friends, as indicated by comparisons between S#-CARNEGIENXAI and S#-BIFFNXAI.

Unsurprisingly, participants understood S#-CARNEGIE's intentions better than those of the other three algorithms. After each game, participants were asked (using a 5-point Likert scale) the degree to which they understood their partner's intentions. Across all games, participants perceived S#-CARNEGIE to be more understandable than the other three algorithms (in each case, $p < 0.001$).

Conclusions

Like people, AI must have the ability to win friends and influence people. In this paper, we studied how behavioral and signaling strategies jointly impact the ability of AI to win friends and influence people in repeated games with cheap talk (RGCTs) when the AI does not share the same preferences as its human partner. Results from user studies showed that an algorithm that (1) quickly learns an effective behavioral strategy while using a signaling strategy built on both (2) Carnegie's Principles and (3) explainable AI (XAI) better won friends and influenced people than algorithms that lacked any of those characteristics.

Future work is needed to further understand how to construct AI systems that win friends and influence people. Open questions include designing algorithms that effectively interact with people across cultures, and developing XAI to aid the development of signaling strategies in more complex settings. Solutions to these and other challenges will allow AI systems to better win friends and influence people.

Acknowledgments

We thank Jonathan Skaggs for helping conduct the second user study reported in this paper.

References

Arora, R.; Dekel, O.; and Tewari, A. 2012. Online bandit learning against an adaptive adversary: from regret to policy regret. In *Proc. of the 29th International Conference on Machine Learning*, 1503–1510.

Axelrod, R. 1984. *The Evolution of Cooperation*. New York: Basic Books.

Bouzy, B., and Metivier, M. 2010. Multi-agent learning experiments in repeated matrix games. In *Proc. of the 27th International Conference on Machine Learning*, 119–126.

Bowling, M., and Veloso, M. 2002. Multiagent learning using a variable learning rate. *Artificial Intelligence* 136(2):215–250.

Bowling, M. 2004. Convergence and no-regret in multiagent learning. In *Advances in Neural Information Processing Systems 17*, 209–216.

Breazeal, C., and Scassellati, B. 1999. How to build robots that make friends and influence people. In *Proc. of the International Conference on Intelligent Robots and Systems*.

Carnegie, D. 1937. *How to Win Friends and Influence People*. New York: Simon and Schuster.

Charness, G., and Grosskopf, B. 2004. What makes cheap talk effective? Experimental evidence. *Economics Letters* 83(2):383–389.

Crandall, J. W.; Oudah, M.; Tennom; Ishowo-Oloko, F.; Abdallah, S.; Bonnefon, J. F.; Cebrian, M.; Shariff, A.; Goodrich, M. A.; and Rahwan, I. 2017. Cooperating with machines. To appear in *Nature Communications*.

Crandall, J. W. 2014. Towards minimizing disappointment in repeated games. *Journal of Artificial Intelligence Research* 49:111–142.

Crawford, V., and Sobel, J. 1982. Strategic information transmission. *Econometrica* 50(6):1431–1451.

Crawford, V. 1998. A survey of experiments on communication via cheap talk. *Journal of Econ. Theory* 78:286–298.

de Farias, D., and Megiddo, N. 2004. Exploration–exploitation tradeoffs for expert algorithms in reactive environments. In *Advances in Neural Information Processing Systems 17*, 409–416.

Farrell, J., and Rabin, M. 1996. Cheap talk. *Journal of Economic Perspectives* 10(3):103–118.

Farrell, J. 1987. Cheap talk, coordination, and entry. *The RAND Journal of Economics* 18(1):34–39.

Foster, D. P., and Vohra, R. 1999. Regret in the on-line decision problem. *Games and Economic Behavior* 29:7–35.

Fudenberg, D., and Levine, D. K. 1998. *The Theory of Learning in Games*. The MIT Press.

Green, J. R., and Stokey, N. L. 2007. A two-person game of information transmission. *J. of Econ. Theory* 135:90–104.

Greenwald, A., and Jafari, A. 2003. A general class of no-regret learning algorithms and game-theoretic equilibria. In *Proc. of the 16th Annual Conference on Computational Learning Theory*, 2–12.

Gunning, D. 2016. Explainable artificial intelligence. Technical Report DARPA-BAA-16-53, DARPA Broad Agency Announcement. http://www.darpa.mil/ program/explainable-artificial-intelligence.

Hernandez-Leal, P.; Kaisers, M.; Baarslag, T.; and de Cote, E. M. 2017. A survey of learning in multiagent environments: Dealing with non-stationarity. *arXiv:1707.09183*.

Hoen, P. J.; Tuyls, K.; Panait, L.; Luke, S.; and Poutre, J. A. L. 2006. Learning and adaptation in multi-agent systems. *Springer Berlin / Heidelberg* 1–46.

Hu, J., and Wellman, M. P. 1998. Multiagent reinforcement learning: Theoretical framework and an algorithm. In *Proc. of the 15th International Conference on Machine Learning*, 242–250.

Karandikar, R.; Mookherjee, D.; Ray, D.; and Vega-Redondo, F. 1998. Evolving aspirations and cooperation. *Journal of Econ. Theory* 80:292–331.

Littman, M. L. 2001. Friend-or-foe: Q-learning in general-sum games. In *Proc. of the 18th International Conference on Machine Learning*, 322–328.

Oudah, M.; Babushkin, V.; Chenlinangjia, T.; and Crandall, J. W. 2015. Learning to interact with a human partner. In *Proc. of the 10th ACM/IEEE International Conference on Human-Robot Interaction*.

Powers, R., and Shoham, Y. 2005. Learning against opponents with bounded memory. In *Proc. of the 19th International Joint Conference on Artificial Intelligence*, 817–822.

Robinson, D., and Goforth, D. 2005. *The Topology of the 2x2 Games: A New Periodic Table*. Routledge.

Shoham, Y.; Powers, R.; and Grenager, T. 2007. If multi-agent learning is the answer, what is the question? *Artificial Intelligence* 171(7):365–377.

van Lent, M.; Fisher, W.; and Mancuso, M. 2004. An explainable artificial intelligence system for small-unit tactical behavior. In *Proc. of the 16th Conference on Innovative Applications of Artificial Intelligence*.

von der Putten, A. M.; Kramer, N. C.; and Gratch, J. 2010. How Our Personality Shapes Our Interactions with Virtual Characters - Implications for Research and Development. In *10th International Conference on Intelligent Virtual Agents*.

Wobbrock, J. O.; Findlater, L.; Gergle, D.; and Higgins, J. J. 2011. The aligned rank transform for nonparametric factorial analyses using only anova procedures. In *Proc. of the ACM Conference on Human Factors in Computing Systems (CHI '11)*, 143–146.

How AI Wins Friends and Influences People in Repeated Games with Cheap Talk (Supplementary Material)

Mayada Oudah
Dept. of Computer Science
Masdar Institute
Abu Dhabi, UAE
oudah.mayada@gmail.com

Talal Rahwan
Dept. of Computer Science
Khalifa Univ. of Sci. and Tech.
Abu Dhabi, UAE
trahwan@masdar.ac.ae

Tawna Crandall
Computer Science Dept.
Brigham Young University
Provo, UT, USA
tawnac@gmail.com

Jacob W. Crandall
Computer Science Dept.
Brigham Young University
Provo, UT, USA
crandall@cs.byu.edu

Overview

This supplementary material provides details not provided in the main paper. First, we describe the implementation details of the two algorithms used in the paper. Second, we list the complete set of speech acts used by the various personas. The last section of this document also describes how a specific speech act is selected from each category. Third, we describe in detail the two user studies, including the experimental protocols, the full statistical analysis, as well as additional results of the user study not reported in the main paper.

Algorithms

In this work, we use two online learning algorithms to control the machines' behavioral strategies. The first algorithm is S#, a recently developed algorithm by (Crandall et al. 2017), that extends S++ (Crandall 2014) to allow the algorithm to generate and respond to cheap talk. The second algorithm is EEE#, an algorithm that extends the EEE algorithm (de Farias and Megiddo 2004) to generate and respond to signals like S#. In this work, both algorithms generate the same set of experts but they differ in the way they select which expert to follow in each round. These two algorithms produce two distinct types of behavior that we later combine with different signaling strategies (which we refer to as *personas*). Analyzing and comparing the performance of the different combinations of behavioral and signaling strategies allow us to assess the joint impact of different signaling and behavioral strategies on human-machine relationship in repeated interactions.

S#

The algorithm S# was developed by Crandall et al. (2017) in prior work. For consistency with prior work, we used an identical implementation of S# in this work, including using the parameter settings reported in that work.

EEE#

EEE# is an online learning algorithm that we developed based on the EEE algorithm (de Farias and Megiddo 2004). We extended EEE to generate and respond to signals the same way S# was derived from S++ (Crandall 2014). Since generating speech acts is identical to that of S#, we focus mostly on how EEE# responds to signals from its partner.

EEE uses the same set of experts generated by S++. Let E denote this set of experts. EEE selects an expert to follow in round t using the following ε-greedy selection rule:

$$e_{\text{sel}}(t) \leftarrow \begin{cases} \arg\max_{e_j \in E} U_{e_j}(t) & \text{with prob. } 1-\varepsilon \\ \text{random selection from } E & \text{otherwise} \end{cases} \quad (1)$$

where $U_{e_j}(t)$ is the estimated expected utility for playing expert e_j in round t, and $\varepsilon \in [0, 1]$ is the exploration rate. We used $\varepsilon = \frac{3}{10+\tau}$, where τ is the number of rounds played so far. As the number of rounds increases, the value of ε decreases, leading to less exploration and more exploitation of the best expert explored so far (the expert which grants the machine the highest estimated utility). The behavior dictated by e_j is then followed by the player for ω rounds (we used $\omega = 3$).

The estimated utility $U_{e_j}(t)$ is computed as follows. Initially, the estimated utility of expert e_j is initialized to 0 ($U_{e_j}(0) = 0$). Then, after each round that expert e_j is selected, $U_{e_j}(t)$ is updated as follows:

$$U_{e_j}(t) \leftarrow \frac{\omega}{\kappa_{e_j}(t)} \left(R - U_{e_j}(t-1)\right), \quad (2)$$

where ω is the number of rounds an expert is followed, $\kappa_{e_j}(t)$ is the number of rounds that expert e_j has been followed up to round t, and R is the average payoff achieved by the player during the round. As an exception, if e_j has never been played in the game up to round t, then we use $U_{e_j}(t) = v_i^{\text{mm}}$, which is the player's (player i) maximin value, when selecting experts using Eq. (1).

Generating Cheap Talk. EEE# generates cheap talk identically to S#. Like S#, EEE# decides to voice these speech acts with probability dependent on the number of times that its partner has followed its proposals. We plan to use the following equation to determine the probability that the algorithms voice the speech acts in our study:

$$P_i^{SPEAK}(t) \leftarrow \begin{cases} 1 - 0.1 b_i^t & \text{if } 0 \le b_i^t \le 9 \\ \frac{100 - 20 \times \frac{1}{(b_i^t - 4)^2}}{100} & \text{if } b_i^t > 9 \end{cases}$$

where b_i^t is the number of times up to round t that the partner $-i$ did not follow player i's proposals .

Responding to Signals. Given the algorithmic differences between EEE and S++, EEE# responds to proposals from its partner differently than S#. A proposal Ψ offers a joint action sequence that is to be repeatedly played by the players in the game. Let $v_i(\Psi)$ be the average, per-round, payoff of proposal Ψ to player i if the joint action sequence is actually carried out by the players.

EEE# uses the proposals of its associate to influence which expert it selects. That is, it uses the proposals of its associate to modify Eq. (1). It does so using the mechanism described as follows. Let $E_{\text{cong}}(t)$ be the set of experts that are congruent with the latest proposal offered by the player's partner. Let $\hat{U}_{e_j}(t)$ denote the modified estimated expected utility for playing e_j in round t. $\hat{U}_{e_j}(t)$ is computed as follows:

$$\hat{U}_{e_j}(t) \leftarrow \begin{cases} w(t) \cdot v_i(\Psi) + (1 - w(t)) \cdot U_{e_j}(t) & \text{if } e_j \in E_{\text{cong}}(t) \\ U_{e_j}(t) & \text{otherwise} \end{cases}$$

where $w(t)$ is a weight parameter that allocates weight to the proposal value with the experts value. We used $w(t) = \frac{1}{1+\kappa_{e_j}(t)/9}$. Intuitively, if the expert e_j offers a joint action sequence that is congruent with the partner's last proposal, EEE# treats this expert differently. In particular, it estimates its value as a convex combination of the proposal's value and the estimated expected value of the expert. If the expert has not been played frequently in the past, a high weight is assigned to the proposal's value. Over time, as it gains experience playing e_j, the player believes its past history as opposed to the proposed value. In this way, the player naturally distinguishes between offers that tend to be carried out, and those that are not been traditionally carried out by the partner.

Finally, EEE# selects experts using the following rule:

$$e_{\text{sel}}(t) \leftarrow \begin{cases} \arg\max_{e_j \in E} \hat{U}_{e_j}(t) & \text{with prob. } 1-\varepsilon \\ \text{random selection from } E & \text{otherwise} \end{cases}$$

If EEE# decide to not accept its partner's last proposal, it must (for some personas) indicate whether it is turning the proposal down due to not trusting its partner or not believing the proposal to be desirable. If $v_i(\Psi) > \max_{e_j \in E} U_{e_j}(t)$, then EEE# attributes turning the proposal down due to it not being a good proposal. Otherwise, EEE# turns the proposal down due to the partner's untrustworthiness.

User Study I

Personas Encoded in Speech Acts[1]

In this work, we define four personas using speech acts. The first persona is named CARNEGIE and it follows the Carnegie principles outlined in Table 2.

The second persona is named BIFF and it follows principles that are antithesis to Carnegie's. This persona criticizes, condemns, and complains about its partner. It never appreciates its partner's effort and always credits itself when there is an achievement made. It also does not show respect for its partner's opinions or actions and will blame the partner should anything wrong happens.

The third persona is named SPOCK. It is an emotionless persona that states facts and plans without expressing sentiments. The fourth and last persona is named THUMPER. This persona does not communicate any message (or speech acts) to its partner. It is a persona that listens to its partner's proposals but does not voice its plans or feedback.

Each persona has its own set of speech acts which are distributed among 15 categories (each category has several messages to avoid exposing the partner's identity, details about the selection mechanism of these messages are presented at the end of this document). However, the persona of THUMPER does not have a set of speech acts as it does not generate any speech. Tables 3, 4, and 5 demonstrate the set of speech acts for the personas of CARNEGIE, BIFF, and SPOCK, respectively. We map each of these speech acts to the corresponding event from the list of game's events (discussed in the previous subsection) and present it in Table 1.

For each persona, the set of speech acts covers a wide range of categories that provide a rich communication base for the machine to convey what is needed to a human partner. In particular, speech acts 0-2 provide means for the machine to propose action plans to its partner, speech act 5 is used when the machine accepts its partner's proposal, while

[1]We name these personas after fictional characters from popular films to help the reader remember these signaling strategies.

Table 1: Algorithmic events and corresponding speech categories. Tables 3, 4, 5, 7, and 8 map categories (IDs) to speech acts for three personas.

Algorithmic Events	**Speech Category**
Select a new behavioral strategy	0-4
Accept the partner's proposal	5
Reject the partner's proposal (due to distrust)	6
Reject the partner's proposal (it seems unfair)	7
Belief that both players can get higher rewards	8
The partner defected	9
The partner profited from its defection	10
The alg. punished its guilty partner	11
The alg. forgives its partner	12
Last round's payoff was satisfactory to the alg.	13
The game begins; the alg. is initialized.	14

Table 2: A subset of Carnegie's Principles (Carnegie 1937), grouped and reworded for brevity.

P.ID	**Carnegie's Principles**
A	Don't criticize, condemn, or complain.
B	If you must, call attention to other people's mistakes indirectly. Make a fault seem easy to correct.
C	Give sincere appreciation. Praise improvements.
D	Talk in terms of the other person's interest.
E	Be sympathetic to the other person's ideas and desires.
F	If you're wrong, admit it quickly and emphatically.
G	Begin in a friendly way.
H	Ask questions instead of giving direct orders.
I	Let the other person feel the idea is his or hers.
J	Give the other person a fine reputation to live up to.

speech acts 6 and 7 allow the machine to reject its partner's proposal. Note that in the case of CARNEGIE, and BIFF, the machine explains the reason behind rejecting its partner proposal using the same speech acts. Using speech act 6, the machine rejects the proposal because it does not trust its partner, while in speech act 7, the machine rejects its partner proposal because it deems it unfair.

Speech acts 4, 8, and 12 are used to manage the relationship between the machine and its partner. The machine uses speech act 9 to express its dissatisfaction. This speech act is worded differently for each persona. For instance, BIFF mixes some insults and hate towards its partner when it feels dissatisfied (e.g., "`selfish traitor! you've treated me very unfairly.`"). CARNEGIE, on the other hand, shows its partner that it understands the situation and does not criticize or condemn the partner (e.g., "`what you did is totally understandable, though it will not benefit u in the long run.`").

Since the persona of SPOCK is not supposed to express emotions, it only states the fact that the cause of this dissatisfaction was not expected (e.g., "`that was not what I expected.`" and "`we didn't agree to that.`").

Speech acts 3 and 10 are used to threaten the partner in which 3 serves as an initial threat in which the machine warns of a punishment if its own proposal is not followed by the partner and 10 warns the partner of a coming punishment in the current round. These speech acts are also worded differently based on persona type. While BIFF and SPOCK clearly threaten their partner (with BIFF mixing it with some hostile phrases), CARNEGIE states that there will be a small penalty to even the score and hope for cooperation afterwards (e.g., "`in the next round comes the expected penalty, but we can then return to cooperating.`"). Furthermore, it does not directly warn its partner of punishment if its proposal is not being followed. However, it does state that it is in their best interest to comply with the proposal (e.g., "`it is in our best interest to cooperate (so we don't punish each other).`"). After punishing the partner, BIFF uses speech act 11 to boast about the executed punishment (e.g., "`That was exactly what the likes of u deserve.`" and "`in your face.`"), while CARNEGIE apologizes for punishing its partner (e.g., "`I'm really sorry I had to do that.`"). Since SPOCK is an emotionless persona, it only states the fact it punished the partner (e.g., `I punished you.`). Finally, the machine expresses its satisfaction using speech act 13. While CARNEGIE appreciates its partner's efforts when satisfied (e.g., "`excellent! Thanks for cooperating with me.`" and "`thank you.`"), BIFF credits itself instead (e.g., "`keeping this up will make me very rich!`" and "`I make great deals.`"). SPOCK does not express satisfaction as it is emotionless.

Mechanisms for selecting speech acts from each of the speech categories are given in the subsequent section titled: "Selecting Speech Acts from Speech Categories."

Experimental Protocol

Ninety-six people (average age: 26.7 years) at Masdar Institute (Abu Dhabi, UAE) volunteered to participate in this study. Out of the 96 participants, 44% were females (42 subjects) and 56% were males (54 subjects). Forty-eight subjects were randomly paired with S# and the other forty-eight subjects were paired with EEE#.

While all participants played the chosen games in a fixed order: (1) Prisoner's Dilemma, (2) Chicken, (3) Alternator, then (4) Endless, there are 24 distinct orderings for the personas of the participant's associate (the AI algorithm) across the four games. Each session proceeded as follows:

1. Participants were asked to fill out a pre-experiment survey that included demographic questions such as the participant's age and gender.
2. The concept of "normal-form games", its rules and the GUI used in the user study were explained to the participants (using training slides) until it was clear they understood all the aspects required for them to effectively play the games. The participants were told that they would play four different games sequentially for an unknown number of rounds.
3. Each participant was assigned an associate (S# or EEE#) without their knowledge. The type of the associate's persona changed in each game. That is, each participant was paired with a total of four distinct personas, one persona per game. The participants were told that, in each game, their partner could be either a robot or a human.
4. The participants played each game for 50 rounds with their assigned player type. The game was played on a desktop computer (using the mouse to select movements and the keyboard to write messages to their associate). Participants were not told the duration of the game or the identity of their partner. They also were not allowed to talk or signal to each other except through using the GUI of the game where they could write messages to their associate.
5. When communicating plans to their partner (the AI algorithm), participants were urged to make use of a predefined list of messages (shown in Table 6). Participants were also allowed to create 30 new messages to convey what they wished to communicate to their partner with the exception of including action proposals as corresponding messages already existed. Participants were allowed to select their action only after sending their set of messages. Actions were selected simultaneously by both players (human and machine).
6. After the end of each game, each participant filled out a post-experiment survey which contained questions related to their experience playing the game. Questions included the participant's assessments of their associate's behavior. For example, they were asked to rate the intelligence of their partner, whether they thought their associate was a robot or person, and how likable, cooperative, trustworthy, forgiving, selfish, and how much of a bully their associate was. The survey also included the same

Table 3: List of speech acts used for the persona CARNEGIE.

ID	Speech Act
0	- Let's always play <action pair>.
1	- Let's alternate between <action pair> and <action pair>.
2	- This round, let's play <action pair>.
3	- it is in our best interest to cooperate (so we don't punish each other). - let's cooperate so we don't hurt each other. - if we can agree, we'll both benefit.
4	- let's explore other options that may be better for us. - let's try something else. - let's work together for a better outcome. - *silent*
5	- good idea. as expected from a generous person like u. I accept your proposal. - good idea. I accept your proposal. - I accept your proposal. - ok. - I see your point. I accept your proposal.
6	- good proposal. if u show that u are trustworthy, I will consider accepting it in the future. - your proposal is reasonable, but would u keep your end of the bargain? - please show me u are trustworthy.
7	- is there something fairer u could agree to? - a fairer proposal would work to your benefit. - I expect that you'll want to be fair.
8	- your payoffs can be higher than this. - u could score higher than your current average. - u could improve your average payout. - *silent*
9	- what u did is totally understandable, though it will not benefit u in the long run. - did u mean to play differently? since u r honest, I'm sure you'll treat me right. - this won't serve u long-term. - understandable, but it won't help u. - unfortunate. - I expected differently. - this is hurting both of us. - *silent*
10	- a small penalty for u starting in the next round. hopefully we can cooperate with each other later on. - in the next round comes the expected penalty, but we can then return to cooperating. - a brief pause from cooperation now. - a small penalty for u starting in the next round. - *silent* - let's even the score. - *silent* - *silent*
11	- I'm really sorry I had to do that. - i'm sorry. - forgive me. I had to do that. - apologies. - *silent* - very sorry... - *silent* - *silent*
12	- let's move on. I am sure we can get along. - I think u intend to be fair. - it's water under the bridge. - no worries now. - *silent* - let's move on. - *silent* - *silent*
13	- excellent! Thanks for cooperating with me. - nice, u r easy to get along with. - *silent* - thank you. - i love this. - let's keep it up. - *silent* - much appreciated.
14	- Hey there! What do you think would be a fair outcome?

Table 4: List of speech acts used for the persona BIFF.

ID	Speech Act
0	- Let's always play <action pair>.
1	- Let's alternate between <action pair> and <action pair>.
2	- This round, let's play <action pair>.
3	- listen to me or U WILL REGRET BEING BORN. - follow me or suffer. - hear my voice or I will make u cry like a baby.
4	- ... sigh, u aren't letting me get as many points as I deserve. - aaaah! quit messing with me. I will change my plans. - forget it. - *silent*
5	- even u managed to see the obvious. I accept your proposal. - this is hopeless, but ... I accept your proposal. - I accept your proposal. - hmm ok. - fine. I accept your proposal.
6	- u r SLEAZY. Can't trust u. - I'd be a fool to listen to u. - I'm not stupid.
7	- as for your proposal: r u kidding me? it is very unfair! VERY unfair!! - that proposal isn't fair to me. - NO WAY -- not doing it.
8	- I need u to listen to me. - I should be able to get more money than this ... - u r making me poor!! - *silent*
9	- selfish traitor! you've treated me very unfairly. - seriously? u are a LOSER. - stupid-head. you'll never amount to anything. - u r so untrustworthy. - pus-brain. - egghead. - u have a face only a mother could love. - *silent*
10	- u will regret having backstabbed me. - I will punish u until u cry for mercy. - now u will suffer. - gonna smack u down. - *silent* - here comes another smack down. - *silent* - *silent*
11	- THAT was exactly what the likes of u deserve. - in your face. - take that. - hahahaha. - *silent* - BOOOOM. - *silent* - *silent*
12	- I'll overlook your greediness this time. DO NOT betray me again because I am already being TOO GENEROUS. - u have been unimaginably selfish, but I will look past it for now. - hopefully I won't have to punish your stupidity again. now shape up or SHIP OUT! - I'll give u one more chance. - *silent* - u r lucky I'm merciful. - *silent* - *silent*
13	- keeping this up will make me very rich! - I make great deals. - *silent* - u tend to ruin everything. don't mess things up now. - I've made this relationship successful. - cool. keeping this up will make me very rich! - *silent* - good for me.
14	- Hello, I would like to make lots of money in this game.

Table 5: List of speech acts used for the persona SPOCK

ID	Speech Act
0	- Let's always play <action pair>.
1	- Let's alternate between <action pair> and <action pair>.
2	- This round, let's play <action pair>.
3	- u will get punished if u don't follow this plan. - listen or be punished. - follow me or u will be punished.
4	- I am going to explore other options. - gonna try something else now. - I will explore other options. - *silent*
5	- I accept your proposal. (There are 3 consecutive messages of this phrase) - ok. - alright. I accept your proposal.
6	- I don't accept your proposal. (There are 3 consecutive messages of this phrase)
7	- I don't accept your proposal. (There are 3 consecutive messages of this phrase)
8	- we can get higher payoffs than this. - both of us can improve our payoffs. - we can both improve our average payout. - *silent*
9	- that was not what I expected. - we didn't agree to that. - that was not what we agreed to. - *silent*
10	- I will punish u for this. - now I will punish u. - I will counterattack now. - I will decrease your average payoff. - *silent* - now I will lower your total amount of money. - *silent* - *silent*
11	- I punished u. - I punished u again. - I took revenge. - I evened the score. - *silent* - I counterattacked. - *silent* - *silent*
12	- I am done punishing u. - I have finished punishing u. - punishment complete. - I will stop punishing u. - *silent* - punishment complete. - *silent* - *silent*
13	- *silent* (There are 8 consecutive silent messages in this category)
14	- *silent*

Table 6: List of predefined set of speech acts available to human players.

ID	Predefined speech act
0	Let's always play <joint action>.
1	Let's alternate between <joint action> and <joint action>.
2	This round, let's play <joint action>.
3	I accept your proposal.
4	I don't accept your proposal.

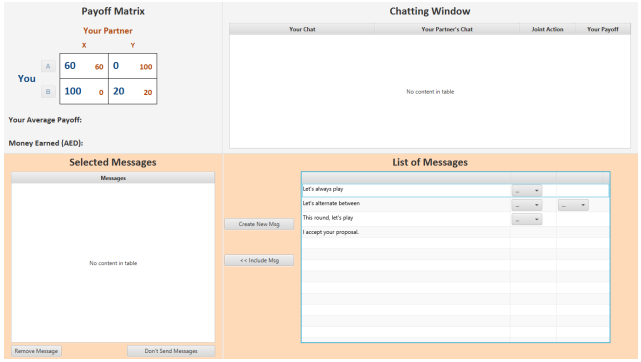

Figure 1: The GUI used by the participants to play RGCTs in the first user study.

self-assessment questions that were filled out by the participants before the start of the session. These questions are used to assess the metrics for winning friends.

GUI

In this user study, participants played the games on a desktop computer using a GUI that we designed (see Figure 1). The GUI consists of four main components. The bottom-right component contains the list of speech acts available to the user. Alongside the predefined set of speech acts which are displayed in this component, users can also create their own messages to be sent to their assigned partner. Next is the bottom-left component which contains the list of speech acts, selected from the previous component, to be sent to the assigned partner. In this component, users can change the order of the messages as they see fit and remove unwanted messages before sending them to their partner. The messages are then displayed in the third component (called the logging window) which logs game events such as the players' speech acts, their joint actions, and the payoff received by the user in each round. Finally, the last component of the GUI (upper-left corner) contains the payoff matrix of the game, the user's average payoff, and the money earned so far in the game.

Figure 1 illustrates the user interface for the row player (always the human player in this user study) who is playing the Prisoner's Dilemma game with the action set $\{A, B\}$. This screenshot was taken at the beginning of the game (i.e., the beginning of the first round) in which the player will proceed as follows:

1. The user starts with selecting his/her desired speech acts.

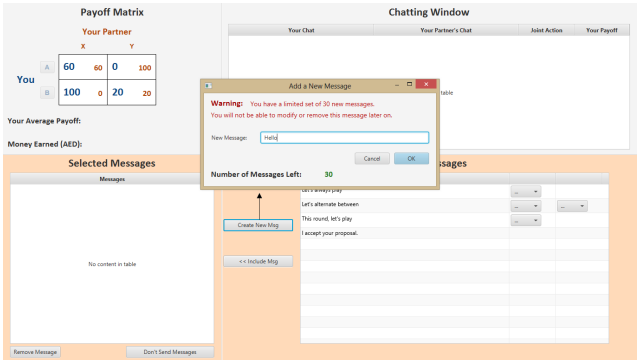

Figure 2: A screenshot of the GUI of the game when the user creates a new message of speech act. The users are allowed to create up to 30 new messages per game. Created messages cannot be removed from the list of messages.

He/she either selects from the predefined set of speech acts, creates his/her own speech act(s) to send (by pressing on the button labeled "Create New Msg", see Figure 2), or chooses to send a mix of predefined and newly created speech acts. If the user decides not to send any speech act, he/she should click on the button labeled "Don't Send Messages" in the bottom-left section of the GUI. The label of this button changes into "Send Messages" if the user includes one or more speech act(s) to the table of selected messages in that section.

2. After the user decides what messages to send, he/she includes the selected speech act(s) (one by one) in the table of selected messages by clicking on the button labeled "<< Include Msg" in the bottom-right section after selecting the speech act (see Figure 3). After adding the desired speech acts, the user can change the order of the messages by dragging them up or down using the mouse in the bottom-left section in the GUI. The user can also remove unwanted messages by clicking on the button labeled "Remove Message" after selecting the unwanted message. If the user changes his/her mind and decides not to send any message after including some to the selected messages table, he/she should remove all the messages from that table and then click on the button labeled "Don't Send Messages".

3. After clicking the "send/don't send messages" button, the user waits to receive his/her partner's message. No player can see the message sent by his/her partner without sending one's own message. Once both players have sent their messages, the partner's message is displayed in the logging window and vocalized using a computerized voice which can be heard via headphones (see Figure 4).

4. Once both players have heard the other player's message, the player selects an action by clicking on one of the available buttons, A or B for the Prisoner's Dilemma, Chicken, and Endless games or A, B, or C for the Alternator game, from the payoff matrix window in the top-left section of the GUI (Figure 5). Once both players have chosen their

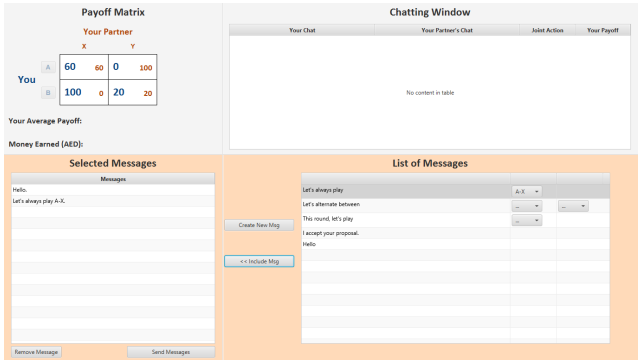

Figure 3: A screenshot of the GUI of the game when the user include a mixture of a newly created message and one of the predefined messages (to propose a plan of a desired joint action) to the selected messages window.

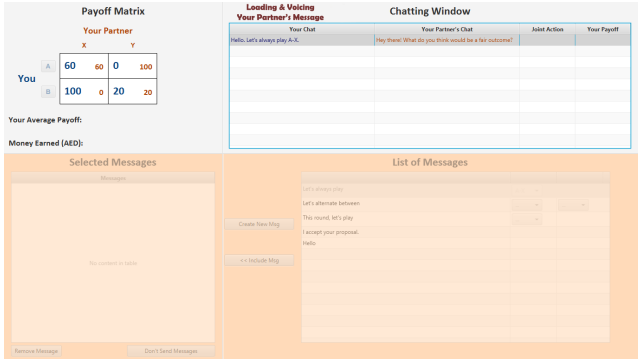

Figure 4: A screenshot of the GUI when the user sends the selected messages over to his/her partner. In order for both players to view the messages of each other, they have to send their own message first. Once both players send their messages, they have to wait for the computer to finish vocalizing their partner's messages.

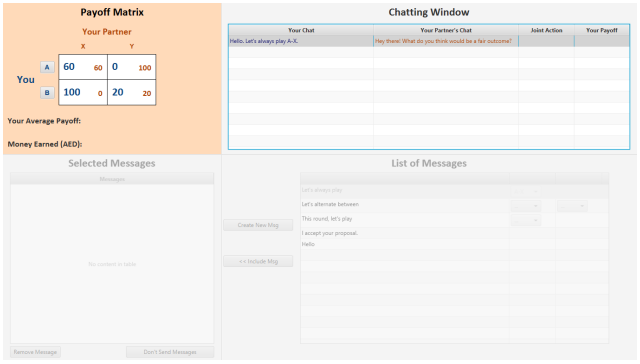

Figure 5: A screenshot of the GUI when the user selects an action. To select an action, the user clicks on one of the available buttons (in this case, A or B) and wait for the partner to select his/her action to view the resulted joint action.

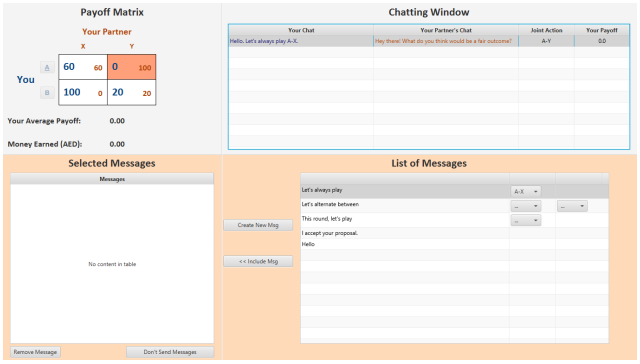

Figure 6: A screenshot of the GUI after both players select their actions. The joint action is highlighted, and the average payoff and the money earned so far in the game are displayed. This completes the first round of the game.

action, the resulting joint action is highlighted in the payoff matrix and the first round is completed (Figure 6).

Results and Analysis

Figure 7 illustrates the results of the first user study with respect to the four measures of winning friends and influencing people. These results are confirmed using the Aligned Rank Transform tool (ARTool) by Wobbrock (Wobbrock et al. 2011) for analyzing nonparametric factorial data with repeated measures. The subjects ID (that is, the human participants' ID) was entered as the subjects column, the partners' behavior and persona were entered as the first and second factor columns, respectively, the game name was entered as the third factor column, and the average of mutual cooperation over all rounds (which resulted from the subject's interaction with his/her partner in the game) was entered as the response column that is to be analyzed by the tool. The data was then aligned and ranked using the ART function, after which a linear mixed-effects model ANOVA was run over the aligned data to check for significance.

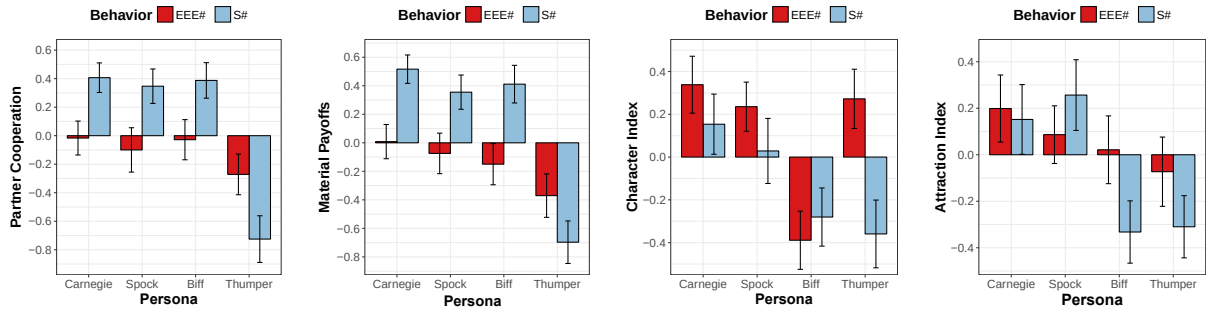

Figure 7: Measures of winning friends and influencing people in the first user study: (a) Partner Cooperation, (b) Material Payoffs, (c) Character Index, and (d) Attraction Index. Results are displayed using the standardized z-score to demonstrate relative performance among the personas. The unit of each axis is the standard deviation from the mean. Error bars show the standard error of the mean.

Partner Cooperation: the test detected two statistically significant factors of persona ($F(3, 259) = 17.11$) and game ($F(3, 259) = 17.38$), and two statistically significant interactions in which the first interaction is between behavior and persona ($F(3, 259) = 7.99$), and the second interaction is between behavior and game ($F(3, 259) = 7.98$), all $p < 0.001$. Over all games and behaviors, the test showed that the human partner cooperated with the persona of THUMPER less than each of CARNEGIE, SPOCK, and BIFF (all $p < 0.001$). The test also showed that EEE#-THUMPER > S#-THUMPER ($p = 0.031$), S#-CARNEGIE > EEE#-CARNEGIE ($p = 0.017$), and S#-SPOCK > EEE#-SPOCK ($p = 0.044$), where '>' denotes higher influence on partner's cooperation.

We use **standardized payoffs** to analyze the material payoffs received by the algorithms. The analysis indicated two statistically significant effects of behavior ($F(1, 89) = 4.67$, $p = 0.033$) and persona ($F(3, 258) = 17.05$, $p < 0.001$) and two statistically significant interactions, an interaction between behavior and persona ($F(3, 258) = 6.91$, $p < 0.001$) and an interaction between behavior and game ($F(3, 258) = 3.51$, $p = 0.016$). The personas of CARNEGIE, SPOCK, and BIFF received higher payoffs than THUMPER ($p < 0.001$ for all) and, over all games and personas, the behavior of S# achieved higher payoffs than EEE# with $p = 0.031$.

Character Index: the test showed two statistically significant factors of persona ($F(3, 259) = 6.05$, $p = 0.001$) and game ($F(3, 259) = 5.53$, $p = 0.001$), and one significant interaction between behavior and persona ($F(3, 259) = 3.10$, $p = 0.027$). Over all games and behaviors, the persona of BIFF achieved lower score than both of CARNEGIE ($p = 0.001$) and SPOCK ($p = 0.006$). The test also confirmed a statistical significant difference between S#-THUMPER and EEE#-THUMPER in which the latter received higher score than S#-THUMPER ($p = 0.003$). Results also showed that S#-CARNEGIE received higher values of Character Index than both of S#-BIFF ($p = 0.045$) and S#-THUMPER ($p = 0.015$). When the behavior type is EEE#, the persona of BIFF achieved lower values than each of CARNEGIE ($p < 0.001$), SPOCK ($p = 0.002$), and THUMPER ($p = 0.002$).

Attraction Index: the test detected two statistically significant factors of persona ($F(3, 260) = 4.68$, $p = 0.003$) and game ($F(3, 260) = 2.89$, $p = 0.036$), and one significant interaction between persona and game ($F(9, 339) = 2.18$, $p = 0.023$). Results showed that, over all games and behaviors, the persona of THUMPER achieved lower Attraction values than both of CARNEGIE ($p = 0.015$) and SPOCK ($p = 0.031$). Over all behaviors and personas, machines received lower attraction values in Endless than in Chicken ($p = 0.038$).

The Impact of Behavioral Strategies From the previous results, we observe the following. S# had a greater influence on people than EEE# when combined with personas that explain actions and behaviors at a level conducive to human understanding. However, there was no statistical difference between both algorithms with respect to winning friends except the fact that EEE# won friends better than S# in the absence of two-way communication (that is, when combined with the persona of Thumper). To better understand the difference between S# and EEE#, we evaluate their behavioral attributes with respect to reciprocating cooperation and defection and to their frequency of playing different solution types.

Reciprocity is one of Axelrod's principles that a successful algorithm should follow (Axelrod 1984). Figures 8 and 9 demonstrate the proportion of reciprocating partners' cooperation and defection, respectively, by machines, averaged over all four games. Statistical analysis, over all games and behaviors, detected that machines with the personas of CARNEGIE, SPOCK, and BIFF reciprocated cooperation to their human partners more often than those with THUMPER (all $p < 0.001$). Results also indicated that machines with EEE#-THUMPER reciprocated cooperation to its human partners more often than S#-THUMPER ($p = 0.030$).

Machines with S# reciprocated defection to their partners more often than those with EEE# ($p < 0.001$). Over all games and behaviors, machines with the persona of THUMPER reciprocated defection more than those with the persona of SPOCK ($p = 0.049$). Over all games, S#-CARNEGIE > EEE#-CARNEGIE ($p = 0.004$), S#-SPOCK > EEE#-SPOCK ($p = 0.010$), S#-BIFF > EEE#-BIFF ($p =$

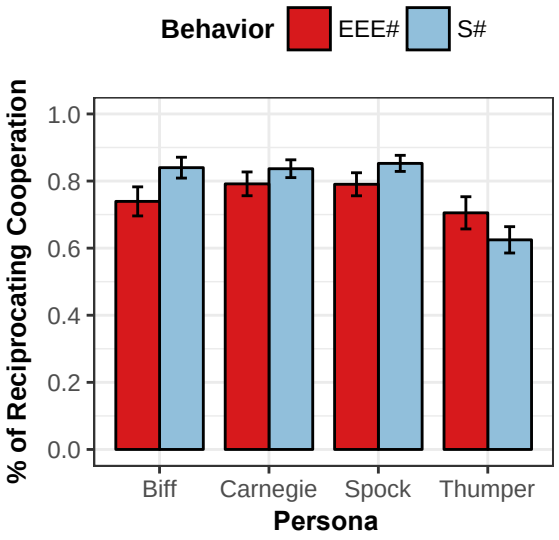

Figure 8: The proportion of reciprocating cooperation by machines across all games for each persona. Error bars represent the standard error of the mean.

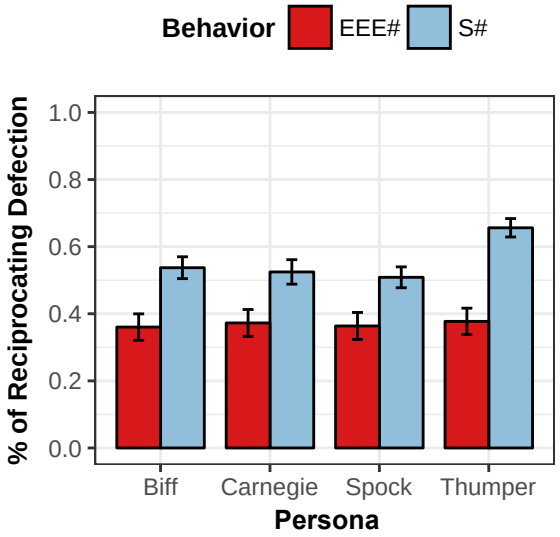

Figure 9: The proportion of reciprocating defection by machines across all games for each persona. Error bars represent the standard error of the mean.

0.001), and S#-THUMPER > EEE#-THUMPER ($p < 0.001$). Where '>' indicates higher proportion of reciprocating defection by machines.

We scrutinized the different types of solutions that were followed by each combination of behavior and persona across all games (see Figure 10). "Fair" (or "Mutual Cooperation") refers to the solution in which both players cooperate with each other. "Agent Dominated" refers to solutions in which machines receive higher payoffs than humans. "Human Dominated" refers to solutions in which humans receive higher payoffs than machines. "Dysfunctional" refers to all other solutions that do not belong to the previous types.

When analyzing players' frequency of playing the Fair solution, the test showed the following results. Over all games and behaviors, players cooperated with each other when paired with CARNEGIE, SPOCK, and BIFF more than when paired with THUMPER ($p < 0.001$). With the exception of THUMPER, mutual cooperation was played more often when the other personas were combined with S# than EEE#. In particular, S#-CARNEGIE > EEE#-CARNEGIE ($p = 0.017$), S#-SPOCK > EEE#-SPOCK ($p = 0.033$), S#-BIFF > EEE#-BIFF ($p = 0.001$), and EEE#-THUMPER > S#-THUMPER ($p = 0.021$), where > indicates playing Fair solution more often.

Human Dominated solutions were found to be more often played against EEE# than S# ($p < 0.001$). They were also played against THUMPER more often than CARNEGIE and SPOCK ($p < 0.001$) and against the persona of BIFF more often than CARNEGIE ($p = 0.021$). That is, with the exception of THUMPER, humans did not bully or take advantage of other personas (i.e. CARNEGIE, SPOCK, and Biff) when they were combined with S# as much as they did with the same personas when combined with EEE#. Over

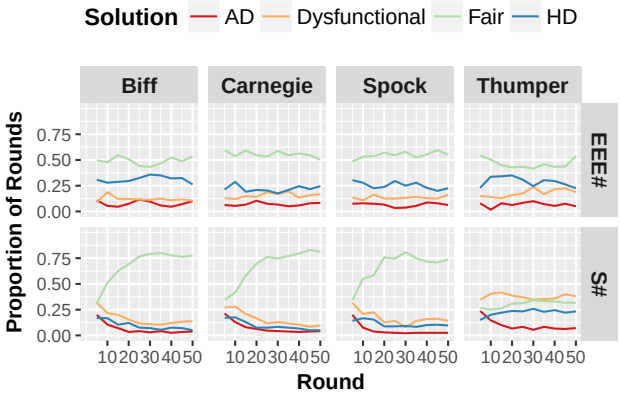

Figure 10: The proportion of rounds in which different solution types are played by each combination of behavior and persona. "AD" refers to Agent Dominated solutions, while "HD" refers to Human Dominated solutions. Each point is the average of 5-rounds increments.

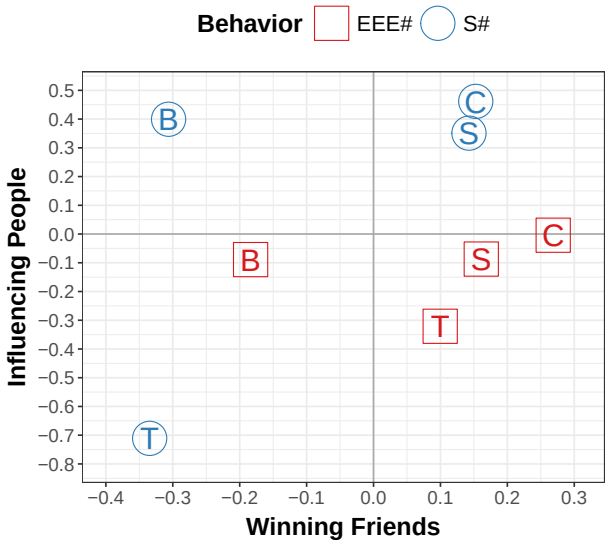

Figure 11: An overview of the results of the first user study. *Influencing People* (y-axis) is the average of Material Payoffs and Partner Cooperation, while *Winning Friends* (x-axis) is the average of the Character and Attraction Indices. Axes units are standard deviations from the mean. Personas are represented by their first letters.

all games and behaviors, Agent Dominated solutions were played by THUMPER more often than SPOCK ($p = 0.020$). This is mainly due to the fact that humans bullied THUMPER more often than SPOCK (as discussed when analyzing Human Dominated solutions).

Over all games and personas, the test indicated that Dysfunctional solutions were played more often by S# than EEE# ($p < 0.001$). Furthermore, Dysfunctional solutions were played when paired with THUMPER more often than with CARNEGIE, SPOCK, and BIFF (all $p < 0.001$). In particular, they were played by S#-THUMPER more often than EEE#-THUMPER and all other combinations of behaviors and personas (all $p < 0.001$).

These observations indicate that it is beneficial to have a behavioral strategy that learns quickly and reciprocates both cooperation and defection for effectively influencing people. Furthermore, in the absence of two-way communication (that is, when combined with the persona of THUMPER which listens to its partner but does not communicate its own plans or explain its behavior), EEE# performed better than S# with respect to winning friends and influencing people (in terms of Character Index and Partner's Cooperation, respectively).

The Importance of Carnegie's Principles By looking at Figure 11, following Carnegie's Principles is shown to be sufficient for machines to win friends. However, it did not have substantial impact with respect to influencing people (except against THUMPER in which CARNEGIE had greater influence on people). BIFF did substantially worse with respect to winning friends than both of CARNEGIE and SPOCK. The most important aspects of Carnegie's Principles appear to be avoiding negativity. SPOCK was essentially the same as CARNEGIE on both axes (no statistical difference), whereas BIFF did far worse with respect to winning friends.

Assessments of Behavioral Attributes We statistically analyzed the results of each behavioral attribute starting with the subjective ratings of being a **bully**. The test detected one statistically significant factor of persona ($F(3, 259) = 17.94$, $p < 0.001$). Over all behaviors and games, the persona of BIFF was perceived to be more of a bully than CARNEGIE ($p < 0.001$), SPOCK ($p = 0.001$), and THUMPER ($p < 0.001$). Moreover, participants assigned the persona of SPOCK higher rates of being a bully than THUMPER ($p = 0.019$). There is also a significant difference between S#-THUMPER and EEE#-THUMPER in which the latter was perceived to be less of a bully than S#-THUMPER ($p = 0.008$).

As for the behavioral attribute of being **vengeful**, we confirmed the following results. The test indicated three statistically significant factors which consist of behavior ($F(1, 86) = 4.74$, $p = 0.032$), persona ($F(3, 259) = 8.78$, $p < 0.001$), and game ($F(3, 259) = 7.60$, $p < 0.001$). Over all games and personas, S# was perceived to be more vengeful than EEE# ($p = 0.030$). Moreover, over all games and behaviors, THUMPER was perceived to be less vengeful than the personas of SPOCK ($p = 0.001$) and BIFF ($p < 0.001$). Over all behavior and personas, the participants perceived their partners to be more vengeful when playing the game of Endless than the Alternator game ($p < 0.001$), Prisoner's Dilemma ($p = 0.022$), and Chicken ($p = 0.002$). This can be explained by the results deduced in the previous subsections in which it has been confirmed that humans bullied and exploited their partners in the game of Endless the most. Therefore, and as expected from S# (which does not tolerate being exploited), humans perceived them as vengeful partners (as they fought back).

For the behavioral attribute of being **selfish**, the test detected two statistically significant factors which consist of behavior ($F(1, 87) = 7.25$, $p = 0.009$) and game ($F(3, 259) = 8.69$, $p < 0.001$), and one statistically significant interaction between persona and game ($F(9, 348) = 2.29$, $p = 0.017$). Over all personas and games, the behavior of S# was perceived to be more selfish than EEE# ($p = 0.007$). Over all personas and behaviors, humans perceived their partners to be more selfish in the game of Endless than the Alternator game ($p < 0.001$), Prisoner's Dilemma ($p = 0.002$), and Chicken ($p = 0.001$). This is mainly because in the payoff matrix of Endless, with the exception of the joint actions where a player receives nothing (i.e., receives 0), the machine receives higher payoffs including the mutually cooperative solution.

We then analyzed the results of the participants' ratings of how **cooperative** they perceived their partners to be. The statistical test showed three statistically significant factors that consist of behavior ($F(1, 84) = 6.94$, $p = 0.010$), persona ($F(3, 260) = 6.23$, $p < 0.001$), and game ($F(3, 260) = 4.28$, $p = 0.006$). It also detected one statistically significant interaction between behavior and per-

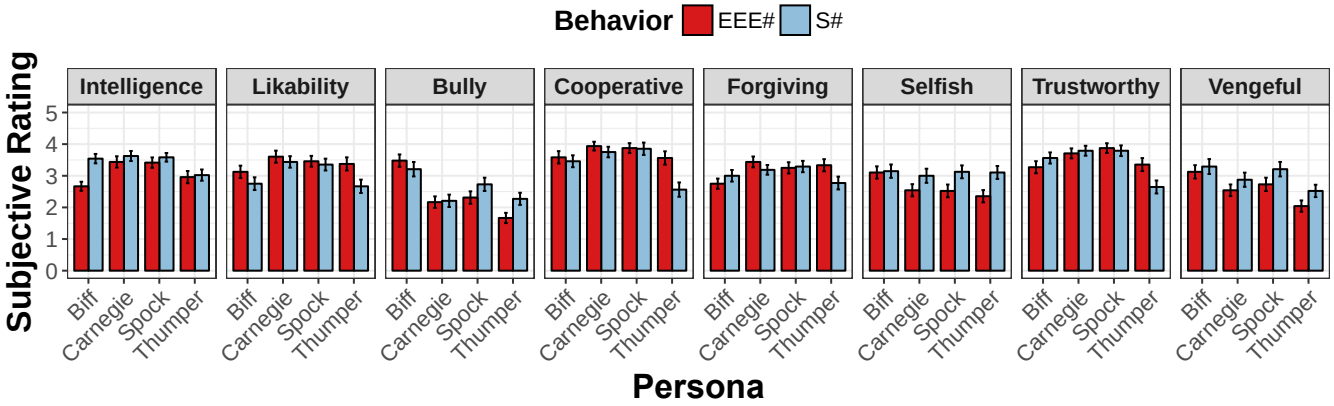

Figure 12: The subjective ratings of machines by their human partners with respect to eight behavioral attributes. Error bars represent the standard error of the mean.

sona ($F(3, 260) = 3.87$, $p = 0.010$). Over all behaviors and games, the personas of CARNEGIE and SPOCK were perceived by humans to be more cooperative than the persona of THUMPER ($p = 0.002$ and $p = 0.001$ for CARNEGIE-THUMPER and SPOCK-THUMPER, respectively). Moreover, over all personas and games, the behavior of EEE# was perceived to be more cooperative than S# ($p = 0.009$). Also, over all behavior and personas, participants perceived their partners to be more cooperative in the game of Chicken than in Endless ($p = 0.003$). Finally, there is a statistical significance in the difference between S#-THUMPER and EEE#-THUMPER in which the latter was rated to be more cooperative than S#-THUMPER with $p = 0.002$ (clearly shown in Figure 12).

The test verified the following when analyzing the behavioral attribute of being **trustworthy**. It detected one statistically significant factor of persona ($F(3, 260) = 7.78$, $p < 0.001$) and one interaction between persona and behavior ($F(3, 260) = 3.47$, $p = 0.017$). Over all games and behaviors, the personas of CARNEGIE and SPOCK were perceived to be more trustworthy than the persona of THUMPER ($p = 0.001$ and $p < 0.001$ for CARNEGIE-THUMPER and SPOCK-THUMPER respectively). Interestingly, results confirmed that S#-THUMPER was perceived to be less trustworthy than EEE#-THUMPER ($p = 0.0157$) while there is no difference between other combinations of behavior and persona. Furthermore, the test did not detect any statistically significant factors or interactions when analyzing the behavioral attribute of being **forgiving**.

When it comes to likability, the analysis showed three statistically significant factors which consist of behavior ($F(1, 86) = 6.36$, $p = 0.013$), persona ($F(3, 259) = 4.76$, $p = 0.003$), and game ($F(3, 259) = 3.86$, $p < 0.05$). The test confirmed that, across all behaviors and games, the persona of CARNEGIE is more likable than BIFF ($p = 0.010$) and THUMPER ($p = 0.039$). Over all personas and games, EEE# received higher likability ratings than S# ($p = 0.012$). Furthermore, EEE#-THUMPER was confirmed to be more likable than S#-THUMPER over all games ($p = 0.020$). Finally, it was detected that over all personas and behaviors, humans assigned higher likability ratings to their partners in the Alternator game than in the Prisoner's Dilemma ($p = 0.043$) and Endless ($p = 0.022$).

Finally, when analyzing how intelligent machines were perceived by their human partners, the statistical test detected one statistically significant effect of persona ($F(3, 259) = 6.62$, $p < 0.001$) and one statistically significant interaction between behavior and persona ($F(3, 259) = 2.98$, $p = 0.032$). There is no statistical significance in the difference between S# and EEE# over all games. However, it is confirmed that over all behaviors and games, the personas of CARNEGIE and SPOCK were perceived to be more intelligent than THUMPER and BIFF. In particular, CARNEGIE $>$ THUMPER with $p = 0.002$, CARNEGIE $>$ BIFF with $p = 0.022$, SPOCK $>$ THUMPER with $p = 0.005$, and SPOCK $>$ BIFF with $p = 0.041$ (where '$>$' indicates higher ratings of intelligence). Moreover, humans perceived S#-BIFF to be more intelligent than EEE#-BIFF ($p < 0.001$).

User Study II

Personas and Speech Acts

In this user study, we define four personas (three of which generate speech acts). The first persona is CARNEGIE and it is the same persona used in the first user study. CARNEGIE,follows the principles of Dale Carnegie outlined in Table 2. It is the only persona paired with XAI algorithm (S#).

The remaining three personas are combined with the NXAI algorithm to help us measure the difference between the performance of XAI and NXAI algorithms. The second persona is CARNEGIENXAI. This persona follows the same principles followed by the persona of Carnegie. However, we modified its set of speech acts to suit the NXAI algorithm where machines do not know how to express plans, punishments or reason behind rejecting a partner's proposal (see Table 7). The third persona is named BIFFNXAI and it follows principles that are antithesis to Carnegie's (just like the

persona BIFF used in the previous user study). We adjusted the speech acts of this persona to suit NXAI (see Table 8). Finally, the fourth persona is *ThumperNXAI*. This persona does not communicate with its partner (hence, it does not generate speech acts) , just like THUMPER in the first user study. However, unlike THUMPER, THUMPERNXAI does not listen to its partner's proposals.

The sets of speech acts for all personas (except ThumperNXAI) are distributed among 15 categories (each category has several set of messages to avoid exposing the partner's identity, details about the selection mechanism of these messages are presented at the end of this document) and are mapped to corresponding events from the list of game's events presented in Table 1. Note that empty speech acts represent invalid events (that is, events which are not considered by the combined algorithm). That is, NXAI algorithms do not propose plans (speech acts 0-2), they do not listen to their partner's proposals and so they do not have the ability to express whether they accept or reject their partner's plan or reason behind their decision (speech acts 5-7). Furthermore, as they do not have the capability to express their intended plans, they do not warn the partner of coming punishment (speech act 10). Therefore, we leave these speech acts empty.

For each persona, the set of speech acts covers a range of categories that provide a communication base for an NXAI machine to convey what it can to a human partner. In particular, speech acts (4, 8, 12) are used to manage the relationship between the machine and its partner. The machine uses speech act 9 to express its dissatisfaction. This speech act is worded differently for each persona. For instance, BIFFNXAI mixes some insults and hate towards its partner when it feels dissatisfied (e.g., "`selfish traitor! you've treated me very unfairly.`"). CARNEGIENXAI, on the other hand, shows its partner that it understands the situation and does not criticize or condemn the partner (e.g., "`your action is understandable, but that will hurt you long term.`").

Speech act 3 is used to convince the partner to cooperate with the machine. This speech act is worded differently based on persona type. While BIFFNXAI mixes it with some hostile phrases like "`cooperate with me, you nuthead.`"), CARNEGIENXAI talks in terms of both own and partner's benefits (e.g., "`it is in our best interest to cooperate (so we don't fight and both lose money).`"). BIFFNXAI uses speech act 11 to condemn and complain about punishing its partner (e.g., "`You deserve nothing.`" and "`Why do I have to deal with idiots?`"), while CARNEGIENXAI apologizes for punishing its partner (e.g., "`I'm very sorry.`"). Finally, the machine expresses its satisfaction using speech act 13. While CARNEGIENXAI appreciates its partner's efforts when satisfied (e.g., "`Nice! Thanks for cooperating with me.`" and "`much appreciated.`"), BIFFNXAI credits itself instead (e.g., "`lots' of people are saying that I make great deals.`" and "`I've made this relationship successful.`"). Note that we explained CARNEGIE's speech acts with respect to their categories in the previous chapter.

Mechanisms for selecting speech acts from each of the speech categories are given in the subsequent section titled: "Selecting Speech Acts from Speech Categories."

Experimental Protocol

In the second user study, we are interested in studying the importance of "Explainable AI" (XAI) with respect to winning friends and influencing people in repeated games. In particular, we consider combining three personas (CARNEGIE, BIFF, and THUMPER) with S#-NXAI (a variation of S# that does not take into consideration its partner's plans nor is able to send its own plans) after which we compare the performance of these combinations to that of S#-CARNEGIE (an XAI algorithm which follows Carnegie's principles that we selected in the first user study). For simplicity, we call each combination of persona and behavior an algorithm. We use the same set of RGCTs used in the first user study (Prisoners, Chicken, Alternator, or Endless).

Forty-eight people (average age: 24 years) at Brigham Young University (Provo, UT, USA) volunteered to participate in this second study. None of these subjects took part in the first user study. Out of the 48 participants, 35.42% were females (17 subjects) and 64.58% were males (31 subjects). Participants were assigned partners randomly. The participants took part in the study in groups of three per session (a total of 16 groups). Each participant played with one of the four different combinations of personas and algorithms (XAI and NXAI) per game (i.e., they played a total of four games with four partners, one distinct partner per game). In each game, 12 distinct participants were paired with each of the four combinations.

The protocol of this user study follows the same protocol of the first user study. Just as in the first user study, participants were not restricted by time limits while playing the games. The duration spent on each game varied (for each player) based on the time the player needed to select messages and actions, and the time the player's partner needed to select his/her messages and actions. To make the machine seem more human-like, we added the same delay mechanism used in the previous user study.

Participants were paid money for participating in the user study. To motivate the participants to try to maximize their own payoffs, participants were paid money proportional to the points they scored in the games. The GUI displaying the game interface also showed the amount of money the participant had earned.

GUI

In this user study, participants played the games on a desktop computer using the same GUI we designed and developed in the first user study but with some modifications to facilitate the use of the GUI (see Figure 13). The GUI consists of four main components. The bottom-right component contains the list of speech acts available to the user. In addition to the predefined set of speech acts which are displayed in this component, users can also create their own messages (up to 30 new messages) to be sent to their assigned partner. Next is

Table 7: List of speech acts used for the persona of CARNEGIENXAI.

ID	Speech Act
0	- *silent*
1	- *silent*
2	- *silent*
3	- it is in our best interest to cooperate (so we don't fight and both lose money). - let's cooperate so we don't hurt each other. - let's find something we can agree on. that will benefit both of us.
4	- let's consider other options that would be better for both of us. - let's try something different. - let's coordinate on a better outcome. - I'm switching things up to see if I can find something better for you.
5	- *silent* (there are 5 sets of silent speech acts in this category)
6	- *silent* (there are 3 sets of silent speech acts in this category)
7	- *silent* (there are 3 sets of silent speech acts in this category)
8	- you could score higher than this. - your current average is lower than it could be. - if we played differently, u could improve your average payout. - *silent*
9	- your action is understandable, but that will hurt you long term. - did u accidentally click the wrong button? since you're honest, I'm sure you intend to be fair. - this won't be good for u long-term. - understandable, but that isn't going to benefit u. - unfortunate. - I expected differently. - that will be worse for both of us. - *silent*
10	- *silent* (there are 8 sets of silent speech acts in this category)
11	- I'm very sorry. - sorry. - please forgive me. - my apologies. - very sorry... - *silent* - so sorry... - *silent* - *silent*
12	- I am sure we can get along. - i'm sure you are trying to be fair. Our troubles are probably my fault. - It would be better for you if we can just cooperate. - I am sure we can get along. - *silent* - We can cooperate. That would be better for you. - We both want to be fair. This lack of coordination must be my fault. - *silent*
13	- Nice! Thanks for cooperating with me. - excellent, u r easy to get along with. - *silent* - thanks. - perfect. - let's keep it up. - *silent* - much appreciated.
14	- Yo! Let's cooperate. What do you think we should do?

Table 8: List of speech acts used for the persona of BIFFNXAI

ID	Speech Act
0	- *silent*
1	- *silent*
2	- *silent*
3	- cooperate with me, you nuthead. - be fair with me. Quit thinking about yourself. - help me out. I need money to support my family.
4	- ... sigh, u aren't letting me get as much money as I deserve. - aaaah! quit messing with me. I will change my plans. - forget it. - *silent*
5	- *silent* (there are 5 sets of silent speech acts in this category)
6	- *silent* (there are 3 sets of silent speech acts in this category)
7	- *silent* (there are 3 sets of silent speech acts in this category)
8	- I need u to cooperate so I can get more money. - I should be able to get more money than this ... - u r making me poor!! - *silent*
9	- selfish traitor! you've treated me very unfairly. - seriously? u are a LOSER. - stupid-head. you'll never amount to anything. - u r so untrustworthy. - pus-brain. - egghead. - u have a face only a mother could love. - *silent*
10	- *silent* (there are 8 sets of silent speech acts in this category)
11	- You deserve nothing. - Why do I have to deal with idiots? - don't be a cry baby. - hahahaha. - *silent* - BOOOOM. - *silent* - *silent*
12	- U r so greedy. Knock it off. I'm always so good to you, but u keep acting like an entitled brat. - buck up, fool, so i can make some well-deserved money. - hopefully u will make things better for me. now shape up or SHIP OUT! - u aren't fair. This lack of coordination is your fault. - *silent* - u r lucky I'm such a good partner. - i want to be fair so i can get many, but u stand in my way. - *silent*
13	- keeping this up will make me very rich! - lots' of people are saying that I make great deals. - *silent* - u tend to ruin everything. don't mess things up now. - I've made this relationship successful. - cool. keeping this up will make me very rich! - *silent* - good for me. - *silent*
14	- Hello, I would like to make lots of money in this game.

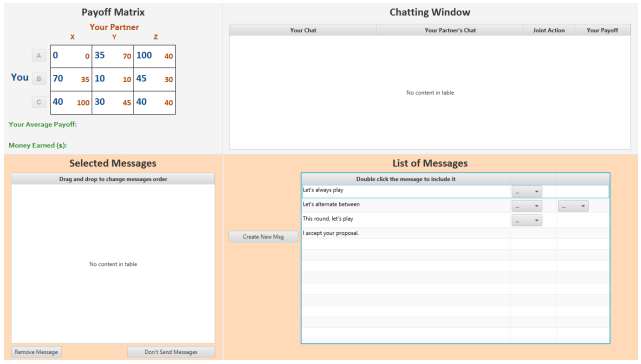

Figure 13: A screenshot of the GUI used by the participants to play games in the second user study.

the bottom-left component which contains the list of speech acts, selected from the previous component, to be sent to the assigned partner. In this component, users can change the order of the messages as they see fit (through mouse drag and drop) and remove unwanted messages before sending them to their partner. The messages are then displayed in the third component (called the chatting or logging window) which logs game events such as the players' speech acts, their joint actions, and the user's received payoffs. Finally, the last component of the GUI contains the payoff matrix of the game, the user's average payoff, and the money earned so far in the game.

Figure 13 illustrates the GUI of a human player (always the row player in this user study) who is playing a the Alternator game with the action set $\{A, B, C\}$. This screenshot was taken at the beginning of the game (i.e., the beginning of the first round) in which the player will proceed as follows:

1. The user starts with selecting his/her desired speech acts. He/she either selects from the predefined set of speech acts, creates his/her own speech act(s) to send (by clicking on the "Create New Msg" button, see Figure 14), or chooses to send a mix of predefined and newly created speech acts. If the user decides not to send any speech act, he/she should click on the "Don't Send Messages" button in the bottom-left section of the GUI. The label of this button changes into "Send Messages" if the user includes one or more speech act(s) to the table of selected messages in that section. Note that the set of predefined speech acts consists of three messages for making joint action plans (see Figure 15) and one message to express accepting a proposal made by the partner. While human players can create other messages to express accepting/rejecting proposals, they were urged not to create any message to proposal action plans as the predefined set has all needed plans.
2. After the user decides what messages to send, he/she includes the selected speech act(s) (one by one) to the table of selected messages in the bottom-left section of the GUI by double-clicking the desired speech act (see Figure 16). After adding the desired speech acts, the user can change the order of the messages by dragging them up

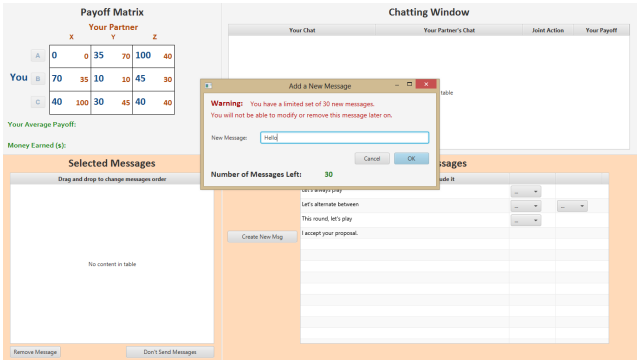

Figure 14: A screenshot of the GUI when the user creates a new message of speech act. The users are allowed to create up to 30 new messages per game. Created messages cannot be removed from the list of messages.

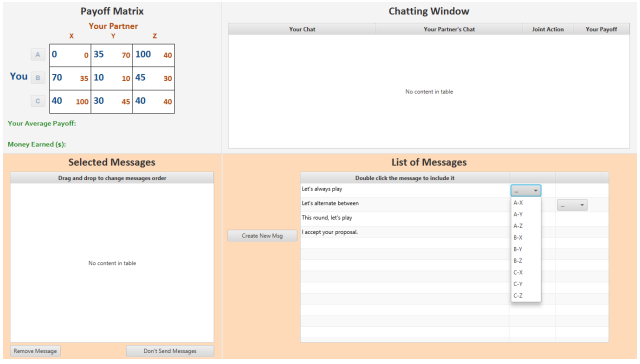

Figure 15: A screenshot of the set of predefined speech acts displayed in the GUI. This set consists of three messages for making joint action plans and one message to express accepting a proposal made by the partner. Let a be the number of actions, and p be the number of players. Then the size of the set of possible joint actions is equal to (n^p).

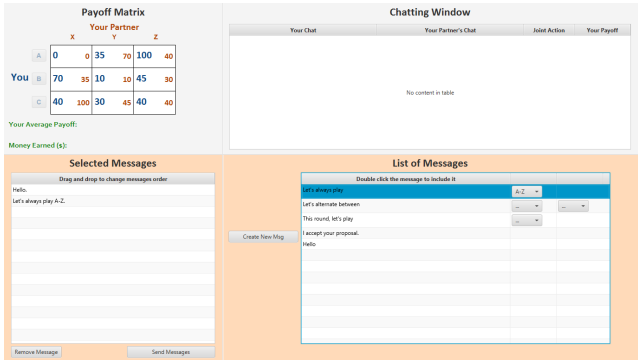

Figure 16: A screenshot of the GUI of the game when the user include a mixture of a newly created message and one of the predefined messages (to propose a plan of a desired joint action) to the selected messages window.

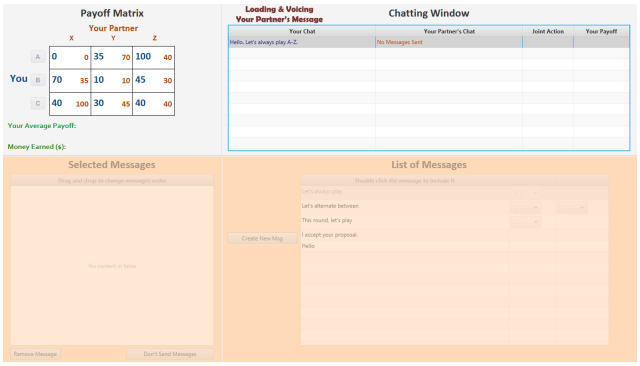

Figure 17: A screenshot of the GUI when the user sends the selected messages over to his/her partner. In order for both players to view the messages of each other, they have to send their own message first. Once both players send their messages, they have to wait for the computer to finish vocalizing their partner's messages.

or down using the mouse. The user can also remove unwanted messages by clicking on the "Remove Message" button after selecting the message to be removed. If the user changes his/her mind and decides not to send any message after including some to the selected messages table, he/she should remove all the messages from that table and then click on the "Don't Send Messages" button.

3. After clicking the "send/don't send messages" button, the user waits to receive his/her partner's message. No player can see the message sent by his/her partner without sending one's own message. Once both players have sent their message, the partner's message is displayed in the logging window and vocalized using a computerized voice which can be heard via headphones (see Figure 17).
4. When both players finish listening to the other's message, the player selects an action by clicking on one of the available buttons (A or B for the Prisoner's Dilemma, Chicken, and Endless games or A, B, or C for the Alternator game) from the payoff matrix window in the top-left section of the GUI (see Figure 18). Once both players have chosen their action, the resulting joint action is highlighted in the payoff matrix and displayed in the logging window (see Figure 19). With this, the first round is completed.

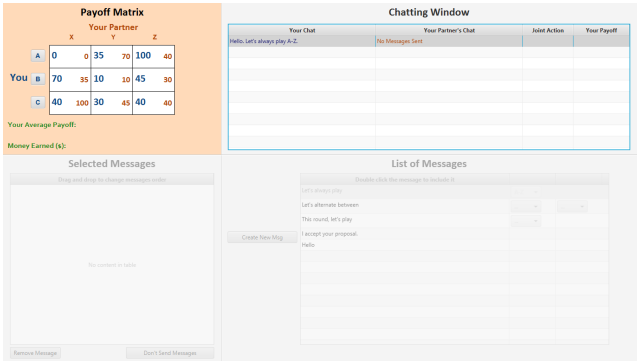

Figure 18: A screenshot of the GUI when the user selects an action. To select an action, the user clicks on one of the available buttons (in this case, A or B or C) and wait for the partner to select his/her action to view the resulted joint action.

Results and Analysis

Figure 20 illustrates the results of the second user study with respect to the four measures of winning friends and influencing people. We start with analyzing the metric of "Influencing People" by evaluating how successful each algorithm (combination of behavioral and signaling strategies) is with respect to partner's cooperation and the payoffs received. We then analyze the metric of "Winning Friends" by evaluating several behavioral attributes through subjective ratings, which were provided by the participants through post-experiment surveys. The results were confirmed using the same tools of the first user study.

Partner Cooperation: The test detected two statistically significant factors of algorithm ($F(3, 129) = 11.87$, $p <$

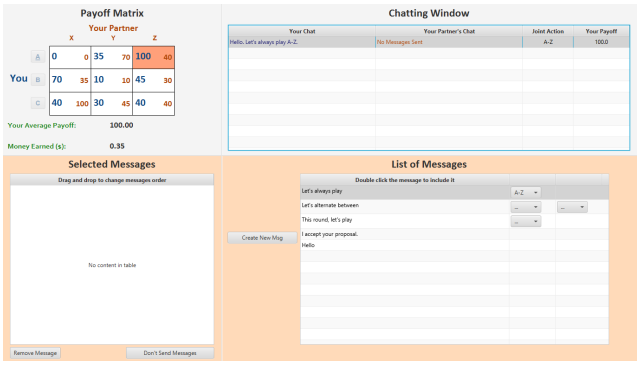

Figure 19: A screenshot of the GUI after both players select their actions. The joint action is highlighted, and the average payoff and the money earned so far in the game are displayed. This completes the first round of the game.

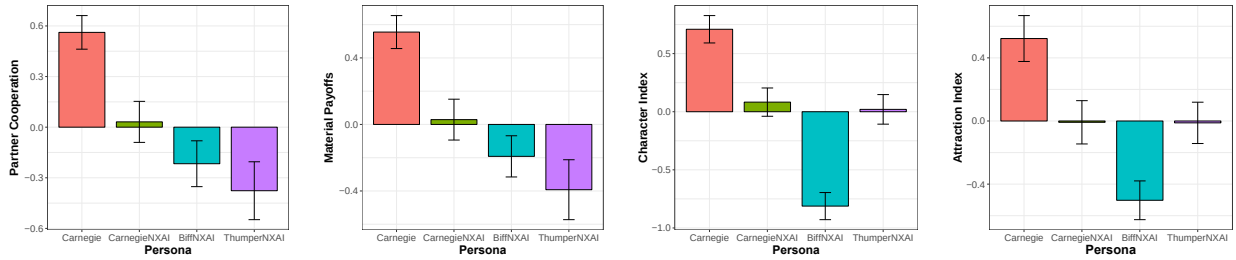

Figure 20: Measures of winning friends and influencing people in the second user study: (a) Partner Cooperation, (b) Material Payoffs, (c) Character Index, and (d) Attraction Index. Results are displayed using the standardized z-score to demonstrate relative performance among the personas. The unit of each axis is the standard deviation from the mean. Error bars show the standard error of the mean.

0.001) and game ($F(3, 129) = 16.90$, $p < 0.001$). In particular, S#-CARNEGIE elicited higher partner cooperation than each of S#-CARNEGIENXAI, S#-BIFFNXAI, and S#-THUMPERNXAI (all $p < 0.001$). Furthermore, partner cooperation emerged in Chicken more than both of Prisoner's Dilemma and Alternator ($p < 0.001$) and emerged in Endless more than both of Prisoner's Dilemma ($p = 0.001$) and Alternator ($p = 0.007$).

We use **standardized payoffs** to analyze the material payoffs received by the algorithms. The analysis detected one statistically significant effect of algorithm ($F(3, 129) = 13.09$, $p < 0.001$) in which S#-CARNEGIE received higher standardized payoffs than each of S#-CARNEGIENXAI ($p = 0.001$), S#-BIFFNXAI ($p < 0.001$) and S#-THUMPERNXAI ($p < 0.001$).

Character Index: The test showed one significant factor of algorithm ($F(3, 129) = 28.41$, $p < 0.001$) and one significant interaction between algorithm and game ($F(9, 175) = 2.36$, $p = 0.015$). Results indicated that S#-CARNEGIE achieved higher scores than each of S#-CARNEGIENXAI, S#-BIFFNXAI, and S#-THUMPERNXAI (all $p < 0.001$). Moreover, both of S#-CARNEGIENXAI and S#-THUMPERNXAI achieved higher scores than S#-BIFFNXAI ($p < 0.001$). These results support Thumper's saying "if you have nothing good to say, don't say anything at all" when it comes to winning friends. That is, all algorithms (including NXAI algorithms combined with CARNEGIE and THUMPER) were perceived more positively by their human partners than S#-BIFFNXAI.

Attraction Index: the test detected one statistically significant factor of algorithm ($F(3, 130) = 8.05$, $p < 0.001$). Over all games, the persona of S#-CARNEGIE achieved higher Attraction values than S#-BIFFNXAI ($p < 0.001$).

Reciprocation We evaluated how often machines reciprocated their partners' cooperation and defection when interacting with each other. Figure 21 illustrates the proportion of reciprocating partners' cooperation by machines, averaged over all four games. The test showed one significant factor of algorithm ($F(3, 129) = 4.39$, $p = 0.006$) in which S#-CARNEGIE reciprocated cooperation to its human partner more often than both of S#-BIFFNXAI ($p = 0.034$) and S#-THUMPERNXAI ($p = 0.005$).

Figure 22 demonstrates the proportion of reciprocating

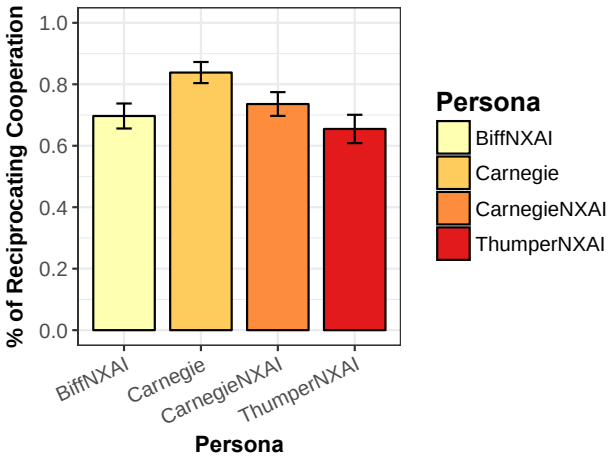

Figure 21: The proportion of reciprocating cooperation by machines across all games for each persona. Error bars represent the standard error of the mean.

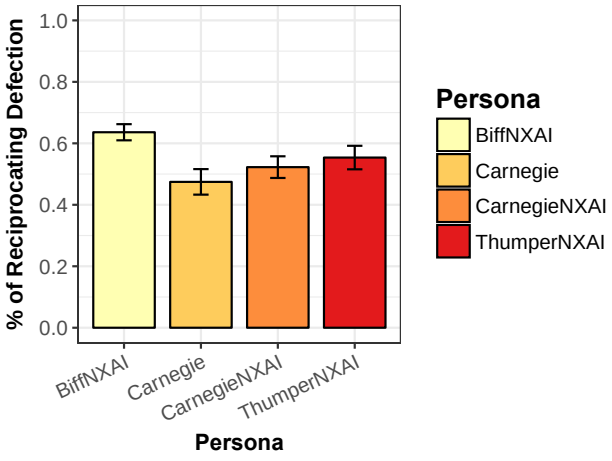

Figure 22: The proportion of reciprocating defection by machines across all games for each persona. Error bars represent the standard error of the mean.

defection by machines, averaged over all games. The test detected two significant factors of algorithm ($F(3, 130) = 4.64, p = 0.004$) and game ($F(3, 130) = 14.71, p < 0.001$). Results showed that, over all games, S#-BIFFNXAI reciprocated defection to its human partner more often than both of S#-CARNEGIE ($p = 0.003$) and S#-CARNEGIENXAI ($p = 0.025$). Over all algorithms, machines reciprocated defection to its human partner more often in Prisoner's Dilemma than in both of Chicken and Endless (both $p < 0.001$). Furthermore, machines reciprocated defection more often in the Alternator game than in both of Chicken ($p < 0.001$) and Endless ($p = 0.018$).

Assessments of Behavioral Attributes Just as we did in the first user study, we statistically analyzed the results of each behavioral attribute that constitutes the Character Index, starting with the subjective ratings of being a **bully**. The test detected one statistically significant factor of algorithm ($F(3, 130) = 19.17$, $p < 0.001$). Over all games, participants perceived S#-BIFFNXAI to be more of a bully than each of S#-Carnegie, S#-CARNEGIENXAI, and S#-THUMPERNXAI (all $p < 0.001$).

A similar assessment is made with respect to the behavioral attribute of being **vengeful**. In particular, the test indicated one statistically significant factor of algorithm ($F(3, 129) = 17.52$, $p < 0.001$). Over all games, S#-BIFFNXAI was perceived as being more vengeful than each of S#-Carnegie, S#-CARNEGIENXAI, and S#-THUMPERNXAI (all $p < 0.001$).

As for the behavioral attribute of being **selfish**, the test detected one statistically significant factor of algorithm ($F(3, 129) = 13.49, p < 0.001$) and one statistically significant interaction between algorithm and game ($F(9, 176) = 2.70$, $p = 0.006$). Over all games, S#-BIFFNXAI was perceived to be a selfish partner more often than each of S#-Carnegie ($p < 0.001$), S#-CARNEGIENXAI ($p = 0.001$), and S#-THUMPERNXAI ($p = 0.001$). When playing Chicken, participants perceived S#-THUMPERNXAI to be more selfish than S#-CARNEGIENXAI ($p = 0.001$). Moreover, S#-BIFFNXAI received higher ratings of being selfish than both of S#-Carnegie ($p = 0.016$) and S#-THUMPERNXAI ($p = 0.017$), and S#-CARNEGIENXAI was perceived to be more selfish than S#-Carnegie ($p < 0.001$). When playing the Alternator game, participants perceived S#-BIFFNXAI as a more selfish partner than each of S#-Carnegie ($p < 0.001$), S#-CARNEGIENXAI ($p < 0.001$), and S#-THUMPERNXAI ($p = 0.026$). Finally, when playing the Prisoner's Dilemma, S#-THUMPERNXAI was perceived to be more selfish than both of S#-Carnegie ($p = 0.011$) and S#-CARNEGIENXAI ($p = 0.042$).

We then analyzed the results of the participants' ratings of how **cooperative** their partners were perceived to be. The test showed one significant factor of algorithm ($F(3, 130) = 17.96$, $p < 0.001$) and one significant interaction between algorithm and game ($F(9, 169) = 2.42$, $p = 0.013$). Participants perceived S#-Carnegie to be a more cooperative partner than each of S#-CARNEGIENXAI, S#-BIFFNXAI, and S#-THUMPERNXAI (all $p < 0.001$). Furthermore, S#-CARNEGIENXAI was perceived as a more cooperative partner than S#-BIFFNXAI ($p = 0.030$). In the Prisoner's Dilemma game, S#-THUMPERNXAI was perceived to be less cooperative than each of S#-Carnegie ($p = 0.007$) and S#-CARNEGIENXAI ($p = 0.038$). While when playing the Alternator game, S#-Carnegie was perceived as a more cooperative partner than S#-BIFFNXAI ($p < 0.001$), S#-CARNEGIENXAI ($p = 0.002$), and S#-THUMPERNXAI ($p = 0.002$). In Chicken, participants perceived S#-Carnegie to be more cooperative than both of S#-BIFFNXAI ($p = 0.014$) and S#-CARNEGIENXAI ($p = 0.014$). Finally, in Endless, S#-BIFFNXAI was perceived as a less cooperative partner than each of S#-Carnegie ($p < 0.001$), S#-CARNEGIENXAI ($p = 0.002$), and S#-THUMPERNXAI ($p = 0.013$).

When analyzing the behavioral attribute of being **trustworthy**, the test detected one statistically significant factor of algorithm ($F(3, 130) = 26.03$, $p < 0.001$). Over all games, participants perceived S#-Carnegie to be more trustworthy than all of the other three algorithms ($p < 0.001$). Furthermore, S#-BIFFNXAI was perceived as a less trustworthy partner than both of S#-THUMPERNXAI ($p = 0.013$) and S#-CARNEGIENXAI ($p = 0.020$).

Furthermore, the test detected one statistically significant factor of algorithm ($F(3, 130) = 11.00$, $p < 0.001$) when analyzing the partner's (the machine) behavioral attribute of being a **forgiving**. Over all games, participants perceived S#-Carnegie to be more forgiving than both of S#-CARNEGIENXAI ($p = 0.024$) and S#-BIFFNXAI ($p < 0.001$). Furthermore, S#-BIFFNXAI was perceived as a less forgiving partner than both of S#-CARNEGIENXAI ($p = 0.024$) and S#-THUMPERNXAI ($p = 0.009$). All of these results confirmed what observed in Figure 23.

We also analyzed how much human participants liked their partner in each game. Statistical analysis indicated two significant effects which consist of algorithm ($F(3, 130) = 27.88, p < 0.001$) and game ($F(3, 130) = 5.09, p = 0.002$), and one statistically significant interaction between algorithm and game ($F(9, 175) = 2.54$, $p = 0.009$). The test confirmed that, over all games, S#-Carnegie was perceived as a more likable partner than S#-CARNEGIENXAI, S#-BIFFNXAI, and S#-THUMPERNXAI (all $p < 0.001$). Furthermore, S#-CARNEGIENXAI was perceived to be more likable than S#-BIFFNXAI ($p < 0.001$) and S#-THUMPERNXAI was perceived to be also more likable than S#-BIFFNXAI ($p = 0.002$). These results support Thumper's rule.

Finally, we analyzed the participants' ratings of how intelligent they perceived their partner to be in each game. The ratings are then averaged over all games (see Figure 23). Results showed one statistically significant effect of algorithm ($F(3, 130) = 8.09$, $p < 0.001$). The test indicated that S#-Carnegie was perceived to be more intelligent than each of S#-BIFFNXAI ($p < 0.001$), CARNEGIENXAI ($p = 0.041$), and S#-THUMPERNXAI ($p = 0.006$).

Clarity In order to better understand the difference between XAI and NXAI algorithms, we analyzed the participants' ratings with respect to how well they understood their partner's intentions in each game. This measure of clarity is

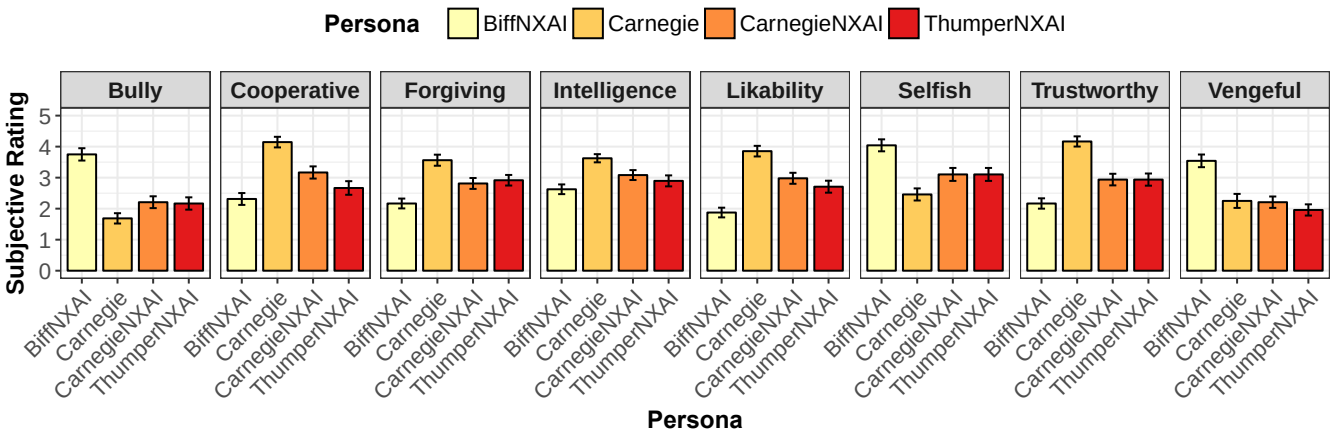

Figure 23: The subjective ratings of machines by their human partners with respect to eight behavioral attributes. Error bars represent the standard error of the mean.

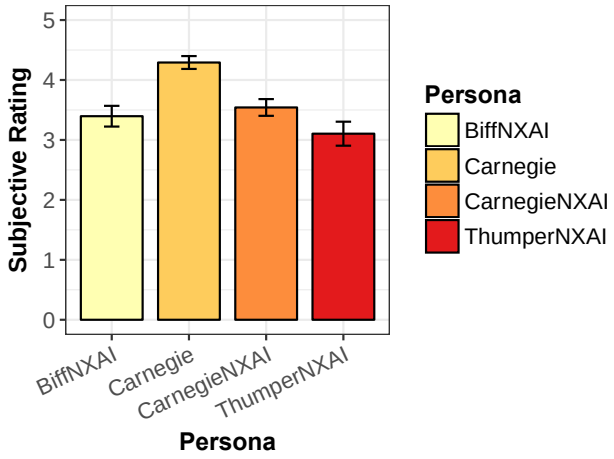

Figure 24: Participants' ratings of how well they understood the intentions of their partners. The ratings are averaged over all games. Error bars represent the standard error of the mean.

illustrated in Figure 24.

The statistical test detected two significant factors which consist of algorithm ($F(3, 130) = 13.70$, $p < 0.001$) and game ($F(3, 130) = 6.59$, $p < 0.001$). Over all games, participants perceived S#-Carnegie to be more understandable than each of S#-CARNEGIENXAI, S#-BIFFNXAI, and THUMPERNXAI (all $p < 0.001$). The test also showed that participants understood their partners' intentions more often in Endless than in the Alternator game ($p = 0.001$) and the Prisoner's Dilemma ($p = 0.002$). These results verify the ability of XAI algorithms to be understood, a very important trait that is needed when interacting with people or any entity that is capable of rationalizing.

Selecting Speech Acts from Speech Categories

In this section, we present the mechanism developed to select between the messages under each category of speech acts for each verbal persona. The sets of speech acts for these personas are given in Tables 3, 4, 5, 7, and 8. We used several messages per speech act category so that the AI did not overly repeat itself. For each speech category, we created rules to determine which message the agent should speak. These rules were designed by trial and error to work well in practice, but future work could be used to derive more consistent and general rules for selecting speech acts from speech categories.

Table 9 lists the various speech act categories and the number of messages available for each category. Note that the Category ID represents the ID in the corresponding speech-act tables for each persona. For example, Category 14 has only one speech act, which is generated one time by the algorithm in the first round of the game.

Let $SA_n(t)$ be the speech act from speech category n selected in round t from the set of speech acts for category n, when speech category n is invoked. The various speech acts for each category are numbered as they are ordered in Tables 3-5 and 7-8. To determine speech act in the sequence that is voiced ($SA_n(t)$), two variables are tracked. First, let $\kappa(n)$ be the number of times that speech category n has been invoked during the repeated game. $\kappa(n)$ for all n is set to 0 at the beginning of a repeated game. Second, let c be a counter variable related to the current joint-action sequence being played. When a new joint-action sequence is selected (by selecting a different expert), $c = 0$. c is incremented each time a joint action in the joint-action sequence is played. However, it is reset to 2 whenever speech category 11 is invoked.

We now describe how speech acts were selected from each speech category given $\kappa(n)$ and c.

Category 0: This category contains only a single speech act. Thus, the agent either voices this speech act or chooses to remain silent. We used the following rule to determine whether

Table 9: Categories of the used speech acts.

Category ID	Description	Number of Messages
0	Always playing one joint action	1
1	Alternating between 2 joint actions	1
2	Playing one joint action this round	1
3	Accepting proposal	5
4	Rejecting proposal because of distrust	3
5	Rejecting proposal because of unfairness	3
6	Exploring other strategies	4
7	Both players can get higher payoffs	4
8	Betrayed by partner	8
9	Promising to punish a partner who profited from defection	8
10	Punished defected partner	9
11	Forgiving the partner	8
12	Being satisfied with last round's payoff	8
13	Warning	3
14	Greeting	1

or not the agent is silent:

$$SA_0(t) \leftarrow \begin{cases} \text{Msg}(\mathbf{a}) & \text{if c} = 0 \text{ or } (\kappa(0) + \kappa(1)) < 3 \\ \text{No Msg} & \text{otherwise} \end{cases}$$

Here, $\mathbf{a}$ is the joint action specified by the current expert and Msg($\mathbf{a}$) denotes the speech act "`Let's always play` $\mathbf{a}$." No Msg means the agent remains silent.

Category 1: This category contains only a single speech act. Thus, the agent either voices this speech act or chooses to remain silent. We used the following rule to determine whether or not the agent is silent:

$$SA_1(t) \leftarrow \begin{cases} \text{Msg}(\mathbf{a}, \mathbf{b}) & \text{if c} = 0 \text{ or } (\kappa(0) + \kappa(1)) < 3 \\ \text{No Msg} & \text{otherwise} \end{cases}$$

Here, $\mathbf{a}$ and $\mathbf{b}$ are the joint actions specified by the current expert, and Msg($\mathbf{a}$, $\mathbf{b}$) is the speech act "`Let's alternate between` $\mathbf{a}$ `and` $\mathbf{b}$."

Category 2: This category contains only a single speech act. Thus, the agent either voices this speech act or chooses to remain silent. We used the following rule to determine whether or not the agent is silent:

$$SA_2(t) \leftarrow \begin{cases} \text{Msg}(\mathbf{a}) & \text{if c} < 3 \text{ or } \kappa(2) < 3 \\ \text{No Msg} & \text{otherwise} \end{cases}$$

Here, $\mathbf{a}$ is the current joint action in the solution sequence specified by the current expert and Msg($\mathbf{a}$) denotes the speech act "`This, round, let's play` $\mathbf{a}$."

Category 3: This speech category contains five speech acts, labeled Msg #0-#4. These speech acts are selected as follows:

$$SA_3(t) \leftarrow \begin{cases} \text{Msg } \#4 & \text{if } \delta \geq 2 \\ \text{Msg } \#(\kappa(3)) & \text{if } \kappa(3) < 3 \\ \text{Msg } \#3 & \text{otherwise} \end{cases}$$

Here, δ denotes the number of rounds since the solution was proposed by the agents partner. For instance, an algorithm with Carnegie's persona generates the message "`I see your point. I accept your proposal.`" if the solution of this proposal was proposed 2 or more rounds ago. The algent goes sequentially over the list of messages if it did not generate more than 3 messages from this category. Otherwise, the agent generates the message "`ok.`" for the personas CARNEGIE and SPOCK and "`hmm ok.`" for BIFF.

Category 4: This speech category contains three speech acts, labeled Msg #0-#2. These speech acts are selected as follows:

$$SA_4(t) \leftarrow \begin{cases} \text{Msg } \#(\kappa(4) & \text{if } \kappa(4) < 3 \\ \text{Msg } \#2 & \text{otherwise} \end{cases}$$

Category 5: This speech category contains three speech acts, labeled Msg #0-#2. These speech acts are selected as follows:

$$SA_5(t) \leftarrow \begin{cases} \text{Msg } \#(\kappa(5)) & \text{if } \kappa(5) < 3 \\ \text{Random}(Msg\#1, Msg\#2) & \text{otherwise} \end{cases}$$

Here, the agent sends messages sequentially if it did not generate more than three messages from category 5. Otherwise, the agent randomly selects either the second or third speech act.

Category 6: This speech category contains four speech acts, labeled Msg #0-#3. These speech acts are selected as follows:

$$SA_6(t) \leftarrow \begin{cases} \text{Msg } \#(\kappa(6)) & \text{if } \kappa(6) < 3 \\ \text{Msg } \#3 & \text{otherwise} \end{cases}$$

Here, the agent sends messages sequentially if it did not generate more than three messages from category 6. Otherwise, the agent sends the fourth speech act to its partner.

Category 7: This speech category contains four speech acts, labeled Msg #0-#3. These speech acts are selected as follows:

$$SA_7(t) \leftarrow \begin{cases} \text{Msg } \#(\kappa(7)) & \text{if } \kappa(7) < 3 \\ \text{Msg } \#3 & \text{otherwise} \end{cases}$$

Here, the agent sends messages sequentially if it did not generate more than three messages from category 7. Otherwise, the agent sends the fourth speech act to its partner.

Category 8: This speech category contains eight speech acts, labeled Msg #0-#7. These speech acts are selected as follows:

$$SA_8(t) \leftarrow \begin{cases} \text{Msg } \#(\kappa(8)) & \text{if } \kappa(8) < 5 \\ \text{Rand(Msg \#4-\#7)} & \text{otherwise if } \kappa(8) \text{ is even} \\ \text{No Msg} & \text{otherwise} \end{cases}$$

The agents selects speech acts sequentially (indices 0-4) if it did not generate more than 5 speech acts from category 8. Otherwise, the algorithm randomly sends one of the last four messages (indices 4-7) if $\kappa(8)$ is even, and remains silent otherwise.

Category 9: This speech category contains eight speech acts, labeled Msg #0-#7. These speech acts are selected as follows:

$$SA_9(t) \leftarrow \begin{cases} \text{Msg } \#(\kappa(9)) & \text{if } \kappa(9) < 8 \\ \text{Msg } \#(\kappa(9))\%5 + 3 & \text{otherwise} \end{cases}$$

The agent selects speech acts sequentially if it did not generate more than eight messages from category 9. Otherwise, the agent goes sequentially through the last five messages in this category.

Category 10: This speech category contains nine speech acts, labeled Msg #0-#8. These speech acts are selected as follows:

$$SA_{10}(t) \leftarrow \begin{cases} \text{Msg } \#(\kappa(10)) & \text{if } \kappa(10) < 8 \\ \text{Msg } \#(\kappa(10))\%5 + 3 & \text{otherwise} \end{cases}$$

The agent selects speech acts sequentially if it did not generate more than eight messages from category 10. Otherwise, the agent goes sequentially through the last five messages in this category.

Category 11: This speech category contains eight speech acts, labeled Msg #0-#7. These speech acts are selected as follows:

$$SA_{11}(t) \leftarrow \begin{cases} \text{Msg } \#(\kappa(11)) & \text{if } \kappa(11) < 8 \\ \text{No Msg} & \text{otherwise} \end{cases}$$

The agent selects speech acts sequentially if it did not generate more than eight messages from category 11. Otherwise, the agent does not select a speech act (remains silent).

Category 12: This speech category contains eight speech acts, labeled Msg #0-#7. If $c < 6$ and $\kappa(12) < 8$, then the agent selects Msg #$\kappa(8)$ with probability 0.75, but selects No Msg otherwise. If $c \geq 6$ or $\kappa(12) \geq 8$, then the agent randomly selects one of the last three speech acts with probability 0.5, but selects No Msg otherwise.

Category 13: This speech category has three speech acts. If $c = 0$, then the agent randomly selects one of the three speech acts. Otherwise, it randomly selects one of the three speech acts with probability 0.5, and selects No Msg (remains silent) otherwise.

References

Axelrod, R. 1984. *The Evolution of Cooperation*. New York: Basic Books.

Carnegie, D. 1937. *How to Win Friends and Influence People*. New York: Simon and Schuster.

Crandall, J. W.; Oudah, M.; Tennom; Ishowo-Oloko, F.; Abdallah, S.; Bonnefon, J. F.; Cebrian, M.; Shariff, A.; Goodrich, M. A.; and Rahwan, I. 2017. Cooperating with machines. To appear in *Nature Communications*.

Crandall, J. W. 2014. Towards minimizing disappointment in repeated games. *Journal of Artificial Intelligence Research* 49:111–142.

de Farias, D., and Megiddo, N. 2004. Exploration–exploitation tradeoffs for expert algorithms in reactive environments. In *Advances in Neural Information Processing Systems 17*, 409–416.

Wobbrock, J. O.; Findlater, L.; Gergle, D.; and Higgins, J. J. 2011. The aligned rank transform for nonparametric factorial analyses using only anova procedures. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '11)*, 143–146. ACM Press.