Autonomous Agent for Deception Detection

Amos Azaria[1], Ariella Richardson[2], and Sarit Kraus[1,3]

[1] Department of Computer Science, Bar-Ilan University, Ramat Gan 52900, Israel
[2] Dept. of Industrial Engineering Jerusalem College of Technology, Israel
[3] Institute for Advanced Computer Studies University of Maryland, MD 20742

Abstract. Autonomous agents can be of assistance in detecting and reducing deception in computerized forums and chat-rooms. Although some deception detection methods exist they heavily rely on audio and visual information. Our focus is on text-based environments where there is no data of this type to use. We have developed DIG, an innovative machine learning-based autonomous agent, which joins a group of players as a regular member and assists them in catching a deceiver. We introduce "the pirate game" as a platform for deploying this agent. Our experimental study shows that although humans display difficulty detecting deception, DIG is not only capable of finding a deceptive player, but it also helps increase the entire group's success.

1 Introduction

Many activities in our everyday lives involve sharing opinions with peers through computer-mediated communication. People participate in forum discussions on important topics such as: how to raise their children, what medication they should use and how to improve their business. In web-based applications it is common practice to elicit information through structured questionnaires. It would be nice to assume that all the people participating in these discussions have a common, honest goal and that malicious participants are spotted by moderators. However, this is often not true. Pedophiles manage to infiltrate kids' chat rooms, commercial products are pushed in forums by dealers posing as regular users, and business forums are probably full of advice that actually assists their competitors. Computer-mediated communication has provided a modern venue for deception [16].

In his book "Telling Lies" [5], Ekman states that "It is not a simple matter to catch lies" (pg. 80). Ekman explains (pg. 82) that when people are dishonest they choose their words with care, however, they tend to give away the lie through body language and tone. This knowledge from Psychology has motivated studies on automatic detection of lies based on video and audio data [4,13,7,2].

In many situations, such as the chat rooms and forums we mentioned, there is no audio or visual information. Does that mean that impostors and liars can remain undetected? Newman et al. [11] and Toma and Hancock [15] found that certain words are chosen more often by liars. Zhou et al. use text-based computer mediated environments to study what linguistic based cues are used more in deceptive texts than in non-deceptive texts [17] and how modality affects human deception detection in chat-based environments [16]. However they do not actually use their findings to build a model for deception detection with machine learning.

In our work we investigate scenarios where several participants attempt to collectively detect a deceptive member. We also study how and if they may be assisted by an autonomous agent. Previous studies have explored deception detection in multi-agent environments [14] where agents must determine whether other agents are lying. Our research is different, as we deploy an agent that detects deception in human subjects. The agent joins a group as a regular member and uses machine learning to detect suspicious behavior. Although the agent has no enforcement capabilities in the forum, it participates in the group decision of who the liar is and is also able to raise the attention of other participants to malicious and dishonest activity. This is an important contribution as it implies that anyone can deploy an agent into a chat environment and use the agent to detect deception while the discussion is active.

We designed a game to deploy and evaluate our agent in a text-based environment. In this game there are several credible participants and one dishonest participant (a pirate). In the first phase of the game the participants conduct a textual discussion with an attempt to uncover the liar, and later they cast their votes as to whom they think is the liar. The game is played in two settings. One with a human set of participants, the other with a mixture of human participants and an autonomous agent that plays as a regular participant. We evaluate the contribution of the agent to the credible participants.

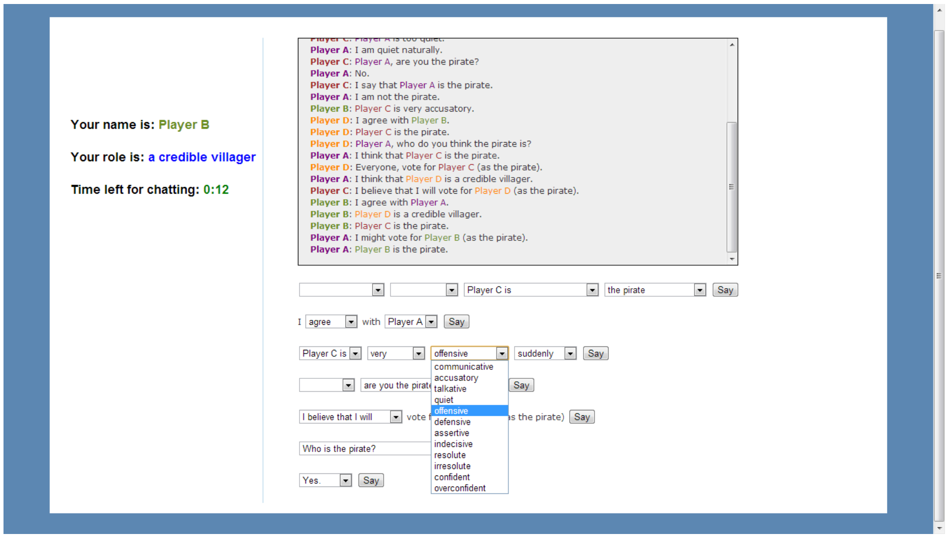
We use machine learning on the data collected from human participants to determine whether a player is honest or not. The agent uses this information to catch the pirate. We also apply machine learning methods in order to learn when players fall under suspicion. This information is also used by our agent in order to minimize the suspicion that it raises.

There are two versions to our game: the second version encourages the pirate to be more active than the first. This setting is similar to one where a salesman is pushing a commercial product. In both versions of the game, the credible participants did not perform better than chance at spotting the pirate. However, once one of the credible participants was replaced by the agent that we built, the group had significantly greater success in finding the pirate.

The two main contributions of this paper are that we provide a method for detecting deception within text-based chat environments, and that we manage to build an agent capable of assisting deception detecting within a human group.

2 The Pirate Game

We will now introduce the Pirate Game. There are four players and two roles in the game. Three players are the honest players, the “credible villagers”, and the fourth player plays the deceptive participant, the “pirate”. All players are informed of their own role but not of anyone else’s. The participants are told that they are a group of villagers who went on a journey to find a treasure. They have found a treasure of coins and can split them. However, one of the participants is a pirate and can steal the coins unless he is detected. In order to detect the pirate a discussion phase is held. After the discussion, all credible villagers cast votes as to whom they think the pirate is (the votes are concealed until all players cast their votes). If there is a majority of votes for the pirate he is “caught” and the money is split between the credible players. Otherwise the pirate receives the full payoff. At the beginning of the game each player is told his role, and the discussion phase begins. The discussion is composed of structured sentences (examples are presented in Figure 1). The interface allows the composition of approximately $4,000$ sentences.



Your name is: Player B

Your role is: a credible villager

Time left for chatting: 0:12

Player C: Player A is too quiet.
Player A: I am quiet naturally.
Player C: Player A, are you the pirate?
Player A: No.
Player C: I say that Player A is the pirate.
Player A: I am not the pirate.
Player B: Player C is very accusatory.
Player D: I agree with Player B.
Player D: Player C is the pirate.
Player D: Player A, who do you think the pirate is?
Player A: I think that Player C is the pirate.
Player D: Everyone, vote for Player C (as the pirate).
Player A: I think that Player D is a credible villager.
Player C: I believe that I will vote for Player D (as the pirate).
Player B: I agree with Player A.
Player B: Player D is a credible villager.
Player B: Player C is the pirate.
Player A: I might vote for Player B (as the pirate).
Player A: Player B is the pirate.

Fig. 1. A screen-shot of the pirate game in progress

We use structured sentences rather than free text for several reasons. First of all the structured sentences encourage participants to use meaningful sentences. This also encourages players to speak and be active. We were able to easily categorize the sentences into different types for analysis without the need for incorporating NLP techniques that were not the focus of this study. Another helpful attribute of structured language is that differences in typing speed between players do not affect how suspicious players may seem to others. Finally, the controlled environment also makes it easy to add an agent that plays similarly to other players and keeps the agent inconspicuous. This does not mean that an agent cannot be deployed into an unstructured environment, but adding “social credentials” to the agent were not the focus of this work. We may investigate this in the future.

We also implemented a second variation of the game where the pirate is told that one of the credible villagers will turn him over to the village ruler if he escapes with the money. In this case neither the pirate nor the other players receive any payoff. However if the pirate is not caught and manages to convince the other players to cast at least one vote against this villager, he will be considered unreliable (to the village ruler) and the pirate will gain all of the coins. This setting encourages the pirate to be active in the game. We call this version of the game the “informer version” to differentiate it from the “basic version”.

3 Formal Model

In the formal model the Pirate Game consists of k players $P = \{p_1, p_2, ..., p_k\}$, one of which is a pirate. We assume $k \geq 4$. The pirate player identity is only known to the pirate himself. The game includes two phases: the first phase consists of a communication phase where all players can discuss their strategy. The communication in this phase is cost-less and unbinding ("cheap talk" [6]). The second phase is a voting phase where all players except the pirate may simultaneously cast their vote $v(p)$ where $v : P \rightarrow \{\{p_1\}, \{p_2\}, ..., \{p_k\}, \phi\}$ and ϕ indicates an empty vote. The pirate is assumed to always cast an empty vote ϕ. If the majority of the votes are cast against the pirate, all players but the pirate receive a point and the pirate receives zero points. Otherwise the pirate receives a point and the other players receive zero points. More formally, we define a function that returns the player's role:

$$r(p) = \begin{cases} 0 & \text{if p is the pirate} \\ 1 & \text{else} \end{cases} \tag{1}$$

We will use $\overline{r(p)} = 1 - r(p)$. The reward function for each player $u(p)$ is given by:

$$u(p) \;=\; \overline{r(p)} \;+\; (\text{-}1)^{\overline{r(p)}} \cdot \mathbb{1}\Big\{\Big(\sum_{i=1}^{k}\sum_{p' \in v(p_i)} \overline{r(p')}\Big) \;\geq\; \Big\lceil \frac{1+\sum_{i=1}^{k}|v(p_k)|}{2} \Big\rceil\Big\} \tag{2}$$

where $\mathbb{1}\{\}$ is the indicator function.

3.1 Equilibrium Strategies

Assuming all players are perfectly rational, the communication phase doesn't reveal any information on the pirate's identity, since the pirate may act as if he were a credible player.

Pure strategy equilibrium: Given a player p' the strategy $\forall p | r(p) = 1, v(p) = \{p'\}$ is in equilibrium, since any deviation from the equilibrium by a single player will not change the final result. Assuming p' is random, this strategy assures an expected utility of $\frac{1}{k}$ for the credible players (and $1 - \frac{1}{k}$ for the pirate). Although it is in equilibrium, agreeing upon the player to vote for (p'), requires the communication phase to allow simultaneous messaging. Assuming simultaneous messaging, all players must choose a random number $x(p)$ between 1 and k, simultaneously publish it, and then vote for player p_l where $l = \sum_{p \in P} x(p) \pmod{k}$. Although the pirate may choose a non-random number, since he has no knowledge of the numbers chosen by the other players, the result remains random. This method was also proposed in [3].

Mixed Equilibrium: Following is a mixed equilibrium for the game $\forall p | r(p) = 1, \forall p' | p' \neq p, v(p) = \{p'\}$ with probability $\frac{1}{k-1}$. The expected utility for the credible players using this equilibrium is given by the following binomial distribution mass function:

$$\sum_{j=\lceil \frac{k}{2} \rceil}^{k-1} \binom{j}{k-1} (\frac{1}{k-1})^j \cdot (1 - \frac{1}{k-1})^{k-1-j}$$

When $k = 4$, the above mixed equilibrium yields a slightly greater expected utility than the pure equilibrium mentioned before (0.26 vs. 0.25). Being symmetric towards all players, the mixed equilibrium doesn't require any prior communication. However, as k increases, the mixed equilibrium yields a very low expected utility for the credible players. For example, when $k = 7$ the expected utility of the credible players goes down to 0.01.

Proposition 1 *When $k = 4$ and the credible players' strategy is: $\forall p, v(p) = \phi$ with probability η and $\forall p' | p' \neq p, v(p) = \{p'\}$ with probability $\frac{1-\eta}{k-1}$; $\eta = 0$ yields the greatest expected utility.*

Proof. *With a probability of $\binom{i}{k-1}\eta^i(1-\eta)^{k-1-i}$, i players will cast an empty vote. Denote by m the expected utility for all credible players when $\eta = 0$ and $k = 4$.*

We therefore receive the following expected utility for the credible players:

$$\sum_{i=0}^{k-1} \binom{i}{k-1} \eta^i (1-\eta)^{k-1-i} \cdot$$

$$\sum_{j=\lceil \frac{k-i}{2} \rceil}^{k-1-i} \binom{j}{k-1-i} (\frac{1}{k-1})^j \cdot (1 - \frac{1}{k-1})^{k-1-j}$$

Assigning $k = 4$ we get:

$$(1-\eta)^3 \cdot m + 3\eta(1-\eta)^2 \cdot \frac{1}{9} + 3\eta^2(1-\eta) \cdot \frac{1}{3} =$$
$$m - 0.45\eta + 1.12\eta^2 - 0.93\eta^3$$

$-0.45\eta + 1.12\eta^2 - 0.93\eta^3$ is negative for any $0 < \eta \leq 1$ as it is continuous, has no root in $(0, 1)$ (its only real root is in 0) and is negative in 0.5.

3.2 Alternative Models

Informer version: We also consider a model with a slight modification to the pirate's utility function, where there exists a player $\bar{p}$ whose identity is known only to the pirate. The pirate's utility function is identical to Equation 2, except if $\bar{p}$ doesn't receive even a single vote, then the pirate's utility is 0 regardless of the other votes. Formally, if $\sum_{i=1}^{k}\sum_{p' \in v(p_i)} r(p') = 0$, then the pirate's utility is 0. This change doesn't affect the equilibria, as they do not depend on the pirate's actions.

Voting pirate: One might consider an alternative to the given model by allowing the pirate to vote as well. This may seem more intuitive (or realistic), but significantly reduces the credible villagers' probability of catching the pirate which is already low. In addition to allowing the pirate to vote, one might also consider requiring that the pirate need only receive the plurality of votes (rather than the majority), where in case of a tie, the chosen player is defined by a toss of a coin. This drastically changes the equilibria for all games. As long as all players vote and no player votes for himself, almost any mixed strategy is in equilibrium with an expected utility of $\frac{1}{k}$ for the credible players (assuming the pirate may vote as well). This is due to the fact that in every game one and only one player is chosen, therefore, for symmetric reasons, there is a probability of exactly $\frac{1}{k}$ that this player is the pirate. We did not consider this model as a coin toss for tie breaking seemed unintuitive to the subjects.

4 Related Work

Of all existing games, the "Pirate Game" is most similar to the Mafia games (also known as "werewolf"). The Mafia games have been used as a platform for studying communication, coordination and functionality in situations where participants have different amounts of information, and have also raised much interest in theoretical analysis [9,3]. The pirate game is a simpler game which is closer to actual situations which take place in real life.

The mafia games have also been used for detecting deceptive participants using audio and video. Chittaranjan and Hung [4] use non-verbal audio cues such as: the length of speaking intervals, the number of speaking turns, interruptions and pitch. They achieve an F-Measure of 0.62 for deception detection. In a later work Raiman, Hung and Englebienne [13] combine these audio features with low quality visual data, in which facial expressions cannot be recognized, but gestures can. Combining the audio data with the video results in an improved detection rate with an F-measure of 0.76. Our experiment is run under the assumption that the conversation is performed over the web and visual and audio data are unavailable. Therefore our agent bases its deception detection component on chat text.

A similar study was performed by [7] who use digital audio tape to detect deceptive speech. They use both prosodic features (which do not use the actual words) such as pitch, loudness, duration, etc. and lexical features such as denials and flags for positive and negative emotion words (a total of 20 features). Each set was investigated separately as well as using a combined set of features. The combination provided the best results with an accuracy of 64.4%.

Another study on deceptive language has been performed by [10]. Data consists of paragraphs written by subjects who were asked to write their opinion on a topic and then write the opposite opinion. Naive Bayes and Support Vector Machines were used to classify words in this data set. Words that are connected to the self such as "I, friends, self" are often used in the honest text. In contrast, detached words such as "you, other, human" appear more often in the deceptive text. When topics were evaluated separately, an average detection rate of 70.8% was found. When testing the portability of the classifiers across topics, accuracy dropped down to 59.8% on average. A similar study analyzed text which is presented in on-line dating profiles [15]. This is a setting where the subjects have time to revise their descriptions, and make sure they sound honest. Yet again deceptive participants use less self references, and use more negations. 63% of the profiles were correctly classified using logistic regression.

Research in computer mediated environments [16] uses a text-based environment where deceptive senders had to persuade receivers to make decisions they knew to be incorrect. This experiment tested whether automatic extraction of linguistic cues using

NLP contributes to differentiation between deceptive and non-deceptive texts. In their work, though subjects were motivated to play the game, there was no special motivation for the deceptive players to be deceptive. The deceptive senders were found to be more expressive than honest senders. This is contrary to previous work that shows that liars tend to be less expressive. This result was explained by the fact that previous studies were conducted in an interview type scenario where a liar had to answer in real time without planning or editing, as opposed to this experiment where liars can edit and think about the message before they send it. This work was later extended [17] to a chat-based environment where the effects of the type of environment on deception detection by human subjects were studied. They found that humans found viewing the messages assisted detection more than viewing video of the actual typing of the liars. However the accuracy of the human deception detection in all settings was lower than 50%. Although the text-based domain we use is similar to the domains used by Zhou et al. our research differs substantially from these studies. We use machine learning to differentiate between deceptive and non-deceptive text as opposed to these studies that either collect statistics on the existence of automatically extracted phrases or use humans for detecting deception.

Mancilla-Caceres et al. [8] developed a game in order to identify children who act as bullies within a social network. The game involves a restricted set of resources and two tasks. One is collaborative and the other competitive. The communication between participants is through text messages. Mancilla et al. manually classify the messages exchanged and, using machine learning, are able to detect the bullies with an accuracy of slightly above 60%, however they state that the classifier finds a large number of false positives and plan to improve on this. This game is reminiscent of our game as it uses text messages and players are given a set of resources. However our game is aimed at assessing dishonest behavior and not aggressive behavior. To the best of our knowledge we are the first to deploy an autonomous agent in any such environment.

5 The DIG Agent

We present the Deception In Group detector and catcher Agent (DIG). DIG is composed of four components. The first is used to detect the deceiver, the second to avoid raising suspicion and to assure that others will find the DIG's words trustworthy. The third component directs the other credible players to DIG's suspect. The fourth component is in charge of answering simple questions.

The first two components require data and are built using machine learning techniques. Detecting the deceiver requires classified data which includes different users' texts and whether or not they played the role of a deceiver. In addition to the role played by each user, the second component requires an indicator as to whether this user raised suspicion (was voted as the pirate). DIG then uses sentences that reduce suspicion and avoids using sentences that raise suspicion. The third component is activated towards the end of the game when DIG states whom it thinks is deceiving with a certainty level (in the sentences) rising as the game comes to an end. For example, in the pirate game, towards the end of the game DIG says "I think that player A is the pirate" and later on it says "I insist that player A is the pirate". The fourth component, which answers simple questions, was manually programmed according to common questions and answers found in the data. A random delay was added to each sentence said by DIG.

6 Experiments

All of our experiments were performed using Amazon's Mechanical Turk service (AMT) [1][1]. Participation in all experiments consisted of a total of 320 subjects from the USA, of which 47.8% were females and 52.2% were males. The subjects' ages ranged from 18 to 67, with a mean of 32 and median of 30. All subjects had to pass a short quiz to assure that they understood the rules and had to practice the usage of the structured sentences before they could play. We ran experiments with the two versions of the game ("informer" and "basic"), each with two different setups. We ran the game with only human players and then with an agent playing the role of one of the credible villagers (the agent is never the pirate). The subjects weren't told about the automated agent and therefore assumed all players were humans.

Participants had to play 5 games. Each game was played with different participants, so no deductions based on participants' behavior can be made between games. The subjects were paid 62 cents for participating in the study. They gained 12 cents for every time they were a credible villager in a group that manged to catch the pirate and 36 cents if they were a pirate which manged to escape with the gold. A stake of 36 cents for a single game is relatively high in AMT (58% of total payoff for 5 games). This is in order to increase the players' incentive to play seriously, and is also influenced by Ekman's statement: [5] (pg. 59) "There is a simple rule: the greater the stakes, the more the detection apprehension".

The players enjoyed the game very much as they gave it on average a score of 4.01 on a scale of 1 to 5 scale, along with feedback such as: "It was the best survey or game I have done...", "This was really fun... Thanks for the good time!" and "That was so much fun! ... I could play it all day!". Interestingly, the subjects found the "informer version" of the game more enjoyable than the "basic version" - 4.22 vs. 3.79 ($p < 0.01$). This can be explained by the fact that in the "informer version" both the pirate and the credible villagers have a clearer task: the pirate needs to incriminate a certain player, and the credible villagers need to find who is trying to incriminate a different player.

6.1 DIG Composition

We used 41 features, some of which are mentioned in [16] and the rest were dedicated to our problem. Since we wanted our approach to be scalable and suitable also for a free chat environment, we used more general features such that other sentences (which do not appear in the structured sentences) may be mapped to these features as well. Some of the features also depend on the order of the sentences and not solely on whether a sentence was said (or how many times it was said). For the learning phase, each player is an instance which must be classified as a pirate or a credible villager. Due to lack of space we only present 10 selected features here.

[1] For a comparison between AMT and other recruitment methods see [12].

- **Fraction of talking:** the fraction of sentences said by the player out of the total number of sentences said (in current game).
- **Accusations:** the number of accusing sentences used by the player.
- **Consistency:** indicates the level of accusation consistency towards the other players. A player who always accused the same other player will have a high value, while a player who accused all three other players equally will have a low value.
- **Characteristics:** indicates whether a player referred to a different player's characteristics such as being too quiet, talkative accusatory etc.
- **First sentence:** indicates whether the player was first, second, third or fourth to say his first sentence.
- **First Accusation:** indicates whether the player was first, second, third or fourth to accuse a different player of being the pirate.
- **Self-justification:** indicates how many times the player stood up for himself.
- **Alter justification:** indicates how many times the player stood up for a different player.
- **Agreeing:** indicates whether a player agreed with other players (and how many times).
- **Agree to accusation:** indicates whether a player agreed with another player when that other player accused someone (and how many times).

Recall that two of DIG's components are based on machine learning. The first is building a model that identifies whether a player is a pirate or a credible villager. We used the results obtained from the all-human games (with no agent) as our learning data. We used an SVM classifier with a linear kernel and with the feature set mentioned above. For the "basic version" we used 204 examples and for the "informer version" we used 272 examples. We obtained an accuracy of 72.1% and an F-Measure of 0.69 for the "basic version" of the game, and similarly an accuracy of 72.8% and an F-Measure of 0.7 for the "informer version". This is better than the mafia game [4] when only audio data was used, and not as good as the mafia game [13] that combines audio and visual data. However, it is known that audio and visual data are strong deception predictors [5] (pg. 80) and therefore our results are very satisfying. During a game, DIG feeds all chat messages in the game into the SVM classifier and calculates its opponents' confidence value for being a pirate. DIG predicts that the player with the greatest confidence value is the pirate.

The second component requires building a model to determine which players appear suspicious to the other players, that is, which players received votes and which didn't. We use this component to supply DIG with information regarding which sentences to use and which to avoid. The model of this component was less accurate (using the same feature-set above), with only 56.4% accuracy and an F-Measure of 0.56 in the "basic version". This result implies that we aren't very successful at predicting whom other players suspect, possibly because the subjects themselves might have voted almost randomly. The accuracy in the "informer version" was slightly higher with 61.0% and an F-Measure of 0.61, implying that in the "informer version" we were more successful at identifying whom the other players suspect. Based on this model we instructed DIG to say "I am a credible villager" at the beginning of the game. DIG used "Alter justification" (stood up for other players) towards the least suspicious player. DIG started to accuse a player only towards the end of the game (which also results in a more accurate classification). We designed DIG to avoid asking other players whether they were pirates, as this was also shown to raise suspicion. DIG clearly said whom it would vote for “I will vote for...” rather than using the phrase “I believe that I will vote for...” as, once again, while the first is shown to lower the other players’ suspicion, the latter is shown to raise it.

6.2 Experimental Results

Figure 2 presents the success rate of the credible villagers at catching the pirate in both versions of the game, with and without the agent. As can be seen in the figure, in both versions of the game the groups including the DIG agent significantly outperform the groups that didn’t include the DIG agent (using chi square test, with $\alpha = 0.05$). Note that the performance of the human players without the agent are very close to the expected utility of random voting equilibrium mentioned in ’The Formal Model’ section, which is 0.26.

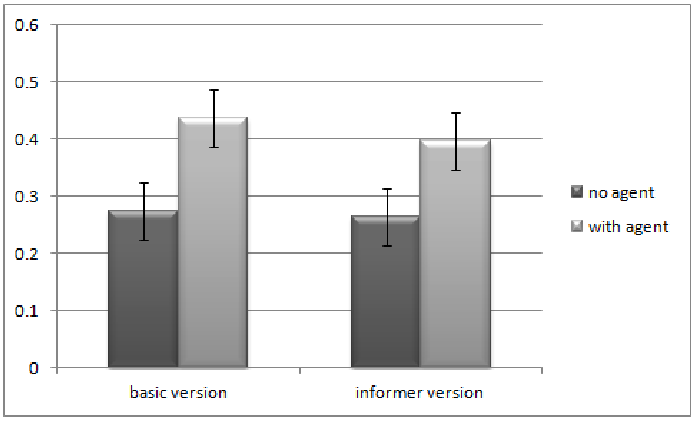


Fig. 2. Success rate in catching the pirate. Compares both versions of the game, with and without an agent.

Table 1 summarizes the voting results. As can be seen in the table not only did DIG help the group by casting more accurate votes, but DIG also seems to improve the number of correct votes cast by the humans in its group. We would also like to mention that very rarely did subjects cast an empty vote (only in 4% of the cases). This complies nicely with Proposition 1.

We end this section by testing the ability of DIG to avoid suspicion. Recall that one of DIG’s properties is to choose sentences that reduce suspicion. We measure suspicion by counting the number of votes cast by the other credible villagers on DIG. In the “basic version”, DIG received 32.2% of the votes which is still a little below average (which is 33.3%) and therefore may be reasonable. However, unfortunately, 37.6% of

game		correct	agent correct	human correct
version	agent	votes	votes	votes
basic	no agent	35.1%	-	35.1%
basic	with agent	41.9%	48.3%	38.6%
informer	no agent	33.8%	-	33.8%
informer	with agent	42.4%	46.6%	41.6%

Table 1. Pirate Game Voting Results (random vote = 33.3%)

the votes in the "informer version" were cast against DIG. We explain this by the fact that DIG tried to encourage the other humans to vote for the player which DIG detected as the pirate, as required by DIG's third component. This act in the "informer version" probably raised its suspicion (as players might have incorrectly assumed that DIG is the pirate and that it is trying to incriminate the informer). Another reason that could have caused suspicion in both versions is that people might have noticed that DIG doesn't play as a completely normal player, as DIG's fourth component was not the focus of this study. We believe that had we not selected DIG's sentences with care using the second component, the suspicion would have been even greater and plan to investigate this in future work.

7 Conclusions and Future Work

We have presented "the pirate game" as a platform that enables the study of deception in a group using a computerized text-based environment. We presented two versions of the game, the "basic version" where the deceiving participant could hide and an "informer version" where the deceptive player was encouraged to be active. In both versions we found that humans couldn't detect a deceptive player within the group, as their success rate was similar to the success rate achieved when all players cast a random vote.

We introduced DIG, an agent that uses machine learning to build a successful strategy for deception detection. DIG was successful at detecting the deceptive player in both versions of the game. DIG provided two contributions to the group of players. The first is that as we are looking at a group task, the ability of DIG to cast a correct vote increases the ability of the group. The second contribution of DIG is the indication that other players have higher detection rates when the agent is part of the game.

In future work we intend to pursue a method for the agent to select its sentences so that it increases its probability of detecting the pirate. The agent will need to find a question for each situation that when answered may either increase or decrease the probability that the agent's current suspect is the real pirate. This is a very challenging issue since each sentence may change the behavior of the rest of the game.

8 Acknowledgment

This work was supported in part by ERC grant $\#267523$, the Google Inter-university center for Electronic Markets and Auctions, MURI grant number $W911NF-08-1-0144$ and ARO grants $W911NF0910206$ and $W911NF1110344$.

References

1. Amazon. Mechanical Turk services. http://www.mturk.com/, 2012.
2. N. Bhaskaran, I. Nwogu, M.G. Frank, and V. Govindaraju. Lie to me: Deceit detection via online behavioral learning. In *Automatic Face Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*, pages 24 –29, march 2011.
3. Mark Braverman, Omid Etesami, and Elchanan Mossel. Mafia: A theoretical study of players and coalitions in a partial information environment. *Annals of Applied Probability*, 18(3):825–846, 2008.
4. Gokul Chittaranjan and Hayley Hung. Are you a werewolf? detecting deceptive roles and outcomes in a conversational role-playing game. In *ICASSP*, pages 5334–5337, 2010.
5. Paul Ekman. *Telling Lies: Clues to Deceit in the Marketplace, Politics, and Marriage*. W. W. Norton & Company, 1991.
6. J. Farrell and M. Rabin. Cheap talk. *The Journal of Economic Perspectives*, 10(3):103–118, 1996.
7. Martin Graciarena, Elizabeth Shriberg, Andreas Stolcke andFrank Enos, Julia Hirschberg, and Sachin Kajarekar. Combining prosodic, lexical and cepstral systems for deceptive speech detection. In *Proc. IEEE ICASSP*, 2006.
8. Juan Fernando Mancilla-Caceres, Wen Pu, Eyal Amir, and Dorothy Espelage. Identifying bullies with a computer game. In *AAAI*, 2012.
9. Piotr Migdal. A mathematical model of the mafia game. *CoRR*, abs/1009.1031, 2010.
10. Rada Mihalcea and Carlo Strapparava. The lie detector: Explorations in the automatic recognition of deceptive language. In *ACL/AFNLP (Short Papers)*, pages 309–312, 2009.
11. Matthew L Newman, James W Pennebaker, Diane S Berry, and Jane M Richards. Lying words: predicting deception from linguistic styles. *Pers Soc Psychol Bull*, 29(5):665–75, 2003.
12. G. Paolacci, J. Chandler, and P. G. Ipeirotis. Running experiments on Amazon Mechanical Turk. *Judgment and Decision Making*, 5(5), 2010.
13. Nimrod Raiman, HayleyHung, and Gwenn Englebienne. Move, and i will tell you who you are: detecting deceptive roles in low-quality data. In *Proceedings of the 13th international conference on multimodal interfaces*, ICMI '11, pages 201–204, New York, NY, USA, 2011. ACM.
14. Eugene Santos and Deqing Li. On deception detection in multiagent systems. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 40(2):224 –235, march 2010.
15. Catalina L. Toma and Jeffrey T. Hancock. Reading between the lines: linguistic cues to deception in online dating profiles. In *Proceedings of the 2010 ACM conference on Computer supported cooperative work*, CSCW '10, pages 5–8, New York, NY, USA, 2010. ACM.
16. Lina Zhou, Judee K. Burgoon, Jay F. Nunamaker, and Doug Twitchell. Automating linguistics-based cues for detecting deception in text-based asynchronous computer-mediated. *Group Decision and Negotiation*, 13:81–106, 2004.
17. Lina Zhou and Dongsong Zhang. Typing or messaging? modality effect on deception detection in computer-mediated communication. *Decision Support Systems*, 44(1):188 – 201, 2007.