# Putting the *Con* in Context: Identifying Deceptive Actors in the Game of Mafia

**Anonymous ACL submission**


## Abstract

While neural networks demonstrate a remarkable ability to model linguistic content, capturing contextual information related to a speaker's conversational role is an open area of research. In this work, we analyze the effect of speaker role on language use through the game of Mafia, in which participants are assigned either an honest or a deceptive role. In addition to building a framework to collect a dataset of Mafia game records, we demonstrate that there are differences in the language produced by players with different roles. We confirm that classification models are able to rank deceptive players as more suspicious than honest ones based only on their use of language. Furthermore, we show that training models on two auxiliary tasks outperforms a standard BERT-based text classification approach. We also present methods for using our trained models to identify features that distinguish between player roles, which could be used to assist players during the Mafia game.


## 1 Introduction

Correct interpretation of language must take into account not only the meaning of utterances, but also characteristics of the speaker and the context in which their utterances are produced. Modeling the impact of this context on language is still challenging for NLP systems. For example, differences in language identification accuracy, speech recognition word error rates, and translation quality have been observed on the basis of attributes such as a speaker's gender, race, dialect, or role (Blodgett and O'Connor, 2017; Tatman and Kasten, 2017; Tatman, 2017; Stanovsky et al., 2019). Moreover, these systems systematically underperform on data generated by those in the minority, having implications for the ethics and fairness of using these technologies.

This work explores language used for deception: a type of speaker context that is particularly challenging to model because it is intentionally hidden by the speaker. To do so, we collect and release a set of records for the game of Mafia, in which each player is assigned either an honest or a deceptive role. Then, we develop models that distinguish players' roles based only on the text of the players' dialog. We describe two auxiliary tasks that improve classification accuracy over a BERT-based text classifier.

The novel contributions of this paper include:

1. A methodology for collecting records of online Mafia games and a dataset collected from 460 human subjects,
2. Three classification models that can distinguish between honest and deceptive players,
3. An approach for identifying features of the game dialog text that can be used to identify deceptive players during the game.

The effectiveness of our classification methods demonstrates that the text of a dialog can be used to identify characteristics of participants automatically, even when those participants are motivated to hide those characteristics by deceiving the listener.

## 2 Background & Related Work

The game of Mafia is particularly well-suited for the goal of determining whether the deceptive participants in a conversation can be identified from the contents of their utterances.

### 2.1 Deception in Language

Humans are a largely collaborative species. However, people sometimes have goals that incentivize them to deceive others. Understanding what cues and interaction styles people adopt when behaving deceptively or seeking to detect deceptive behavior will be crucial to both developing automated detection and a greater understanding of the complex interactions that people use in deception and revelation. Previous work indicates that people struggle with telling apart lies from truth, especially with deceptive statements (Bond Jr and DePaulo, 2006). This raises the question of what strategies deceptive actors use to avoid detection, as well as what strategies honest actors use to discover deceivers.

Deception is a hard topic to study, however, because of its inherent complexity: multiple people with different motivations are trying to evaluate one another, while contending with moral obligations and accusations, over a period of time that involves planning, taking actions, and responding to others' actions. Moreover, there is a distinction between a falsehood, which is a statement that is not true, a lie, which is a statement that the speaker does not believe, and deception, which is the act of convincing another person to hold a false belief. Whereas falsehoods and lies are properties of statements, deceptive intent is a characteristic of the speaker. Therefore, though deceptive speakers may tell falsehoods and lies, they might also provide truthful statements, and vice versa for honest speakers, thus rendering the truth conditions of individual utterances as unreliable indicators of deception. We are interested in how people solve these dual problems of deceiving and detecting deception, which requires a paradigm wherein we can observe all agents' actions and communication while simultaneously knowing agents' underlying incentives and goals. We thus turn to a game with a rich history of deception research: Mafia.

Previous work on detecting deception from linguistic cues has explored scenarios that either mimic or are taken directly from real-world investigations of potentially deceptive actors. Derrick et al. (2013) showed that deceptive parties take longer to formulate responses and use fewer words in the context of chat-based communication. Burgoon et al. (2003) similarly found that deceivers sent briefer chat messages. Fuller et al. (2011) demonstrated the effectiveness of training classifiers to identify deceptive language in relation to crimes, and found that word quantity was a particularly useful feature. Fornaciari and Poesio (2013) also found surface-level features useful in detecting deceptive statements in a criminal context, specifically through the investigation of Italian court documents, while Mihalcea et al. (2013) found that written lies were easier to detect than transcripts of spoken ones. Abouelenien et al. (2014) took a multimodal approach to deception detection, finding that non-contact approaches were able to match or exceed the performance of those that were more invasive.

### 2.2 The Game of Mafia

Researchers have also examined deception in games, focusing on settings such as Diplomacy or negotiation over a set of items (Lewis et al., 2017; Niculae et al., 2015). In addition, there has been some work exploring the effects of biased voting on group decision making (Kearns et al., 2009). The game of Mafia specifically has attracted attention, and researchers have analyzed data from various online game communities. Zhou and Sung (2008) discovered differences between deception across cultural communities by analyzing data from an online Chinese Mafia game, Pak and Zhou (2011) used social network analysis to detect deceivers using the epicmafia.com website, and de Ruiter and Kachergis (2018) collected and trained models on a dataset from the online Mafiascum forum. Researchers have also studied the game of Werewolf, a variant of Mafia. Chittaranjan and Hung (2010) used audio information to classify malicious parties, while Demyanov et al. (2015) used video information. Braverman et al. (2008) and Migdał (2010) developed a mathematical model of the Mafia game, assuming that all votes are cast at random, which allowed them to analyze how mafia and bystander win rates varied with role distribution in a highly controlled version of the game. Bi and Tanaka (2016) showed that under certain conditions, the strategy of mafia pretending to be bystanders is suboptimal.

Most of the deception-oriented games that have been studied provided individual incentives to the players. Mafia allows for the study of patterns of deception that arise when incentives are only at the group level. In addition, whereas using datasets of online Mafia games presents a rich source of deceptive language, the complicated rule sets of games on these forums makes it challenging to isolate specific strategies that participants use to engage in and detect deceptive behavior. In contrast to work using video or audio, we assume that players do not have access to any audiovisual clues about others' roles, thus proposing a more stringent threat-detection model, which we believe is more congruent with the majority of interactions that users have with unverified parties online. Though analyzing mathematical models of Mafia gives insight into certain game mechanics, we wish to focus on the methods that actual players use in order to conceal their own or discover others' roles. This work takes these factors into account by studying a controlled environment that nonetheless supports the use of complex strategies for deceiving and detecting deceptive behavior.

## 3 Dataset

A total of 460 English-speaking participants based in the United States were recruited from Amazon Mechanical Turk using the experiment platform Dallinger[1]. Between 4 and 10 participants were recruited for each Mafia game: 1 to 2 participants were designated mafia, and the rest were bystanders. Forty-four of these Mafia games are included in the analysis. Participants were paid \$2.50 for completing the task, plus bonuses for time spent waiting for other participants to arrive in a chatroom to begin the experiment (waiting was paid at \$5.00/hour).

Upon recruitment, participants were shown a consent form, per IRB approval, followed by an instructional video and accompanying transcript describing how to play the text-based Mafia game (see Appendix). After they completed a quiz demonstrating they understood the information, they entered a waiting room until the desired number of participants was reached. Participants were then assigned a role (mafioso or bystander) and fake name, after which they began playing the game.

The game dynamics were as follows. Each mafia member was aware of the roles of their fellow mafia members and thus, by process of elimination, knew the roles of the bystanders. However, the bystanders did not know the true role of anyone else in the game. The goal of the mafia was to eliminate bystanders until the number of mafia was greater than or equal to that of the bystanders. The goal of the bystanders was to identify and eliminate all of the mafia members. The game proceeded in phases, alternating between nighttime and daytime (Figure 1). During the nighttime, mafia members could secretly communicate to decide on who to eliminate, after which they discretely voted, and the person with the majority vote was eliminated from the game. If there was a tie, one of the people involved in the tie was randomly chosen to be eliminated. During the daytime, everyone was made aware of who was eliminated during the nighttime, and then all players could openly communicate to decide who to eliminate. All the players then voted publicly, and the person with the majority vote was eliminated and announced to be a bystander or mafioso. Thus, during the nighttime mafia could secretly communicate and eliminate anyone, whereas during the daytime mafia could participate in the voting and communication protocols in the same way as bystanders. The game proceeded until there was a winning faction according to the goals described above.

From these experiments, we collected a dataset consisting of both mafia and bystander utterances over the course of each game, as well as the participants' voting behavior. For our analysis, we consider just the daytime dialog in the game, as only the mafia members were able to converse during the nighttime. Figure 2 displays a snippet of the daytime dialog from one Mafia game. As shown here, many utterances are either social interactions (eg. "hi erybody") or discussions about what to do in the game, such as accusations or comments about voting (eg. "I bet it's Mandy...").

[1] http://github.com/dallinger/Dallinger

## 4 Approach

To investigate whether linguistic information can be used to identify players' roles, we train and evaluate classifiers that predict the role of a particular player, which is either mafioso (the deceptive role) or bystander (the honest role). Since we have a small dataset, we chose to fine-tune pre-trained Transformer models rather than train them from scratch (Vaswani et al., 2017). To predict the role for a player $p$, we construct an input representation $r(C, p)$ of the full game dialog $C$ that encodes the player of interest $p$. We develop three approaches which differ in both the dialog representation function $r$ and the modeling approach.

### 4.1 Standard Classification

Our baseline approach uses a standard BERT-based text classifier (Devlin et al., 2018). To classify player $p$ via the full record of the game $C$, let boolean variable $M_p$ be true if $p$ is a mafioso. Let $T_p$ be the concatenation[2] of utterances made by $p$. We train BERT parameters $\theta_M$ to predict $P(M_p|T_p; \theta_M)$. While this model can be used directly to predict the role of a player, we find that

[2] Utterances are concatenated such that there is an end-of-sentence delimiter between them.

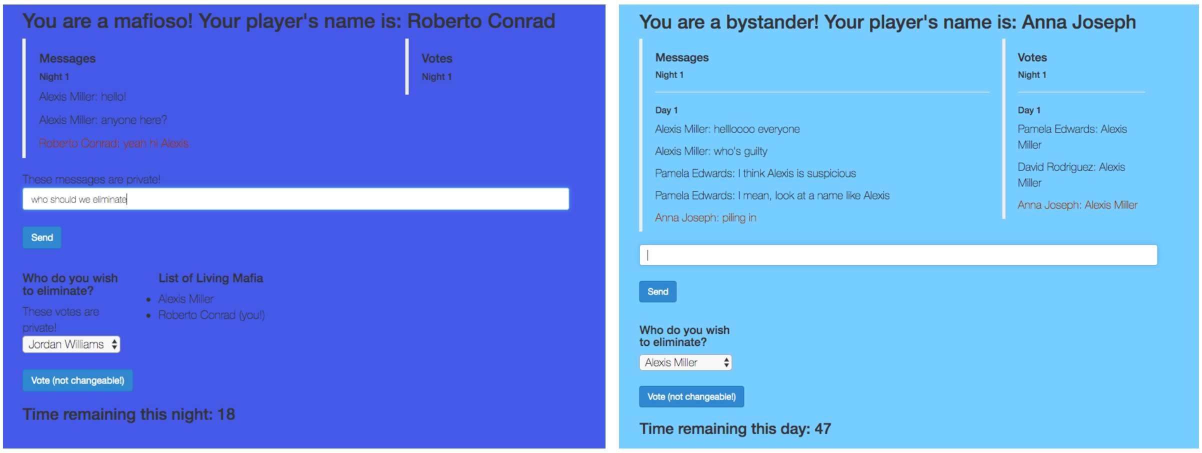


Figure 1: Mafia experiment screenshot during **(left)** first nighttime phase, with participant as a mafioso, and **(right)** first daytime phase, with participant as a bystander (note that mafia messages are not visible to the bystander).

| | creation_time | contents |
|---|---|---|
| **0** | 2018-11-02 21:00:33.658168 | Sarah Bryant: hi erybody |
| **1** | 2018-11-02 21:00:39.856949 | Julie Monroe: I bet it's Mandy. Mandy is an evil name |
| **2** | 2018-11-02 21:00:40.196923 | Mandy Smith: Hello |
| **3** | 2018-11-02 21:00:48.892878 | Mandy Smith: C'mon guy |
| **4** | 2018-11-02 21:00:51.380136 | Mandy Smith: I'm nice |

Figure 2: Example of messages sent between players. *creation_time* is the time at which the message was sent. *contents* consists of the name of the sender, as well as the message, separated by a colon and space.

weighting this prediction by the prior probability $P(M_p)$, which is the fraction of players who are mafia members, improves performance:

$$P(M_p|T_p) = \frac{P(M_p)P(M_p|T_p;\theta_M)}{\sum_{R\in\{M,\neg M\}} P(R_p)P(R_p|T_p;\theta_M)} \quad (1)$$

where $P(\neg M_p) = 1 - P(M_p)$.

This approach, which provides as input to the classifier only the utterances of the player to be classified, outperformed an alternative representation $r(C, p)$ that included the entire record of all utterances by all players.

### 4.2 Auxiliary Tasks

Limiting the input representation $r$ to contain only the speech of the player $p$ being classified is not ideal; correctly interpreting a dialog requires considering all other players' statements as well. We introduce two auxiliary tasks that involve the entire game dialog $C$:

1. Given all of the prior utterances, is a bystander or a mafia member more likely to have produced the current utterance? (*Utterance Classification*)
2. Given all of the prior utterances, what would be the current utterance of a bystander or a mafia member? (*Utterance Generation*)

We develop a BERT-based model for task 1 and fine-tune the GPT-2 language model for task 2 (Radford et al., 2019). Then, we use each of these auxiliary models to classify the role of a particular player $p$ in the game.

#### 4.2.1 Utterance Classification

To classify player $p$ using the auxiliary task of utterance classification, let boolean variable $S_i$ be true if utterance $C_i$ was made by a mafioso (rather than a bystander). Let $C$ be the full record of utterances in the game and $C_{\leq i}$ be the concatenation of all utterances $C_1 \ldots C_i$. We train BERT parameters $\theta_S$ to predict $P(S_i|C_{\leq i};\theta_S)$. Finally, let $I_p$ be the set of indices of utterances by player $p$. $M$ relates to $S$ in that if $M_p$ is true, then $S_i$ is true for all $i \in I_p$. Therefore,

$$P(M_p|C;\theta_S) \propto \prod_{i\in I_p} P(S_i|C_{\leq i};\theta_S).$$

Again, we find that weighting this prediction by the prior $P(M_p)$ improves performance, and we

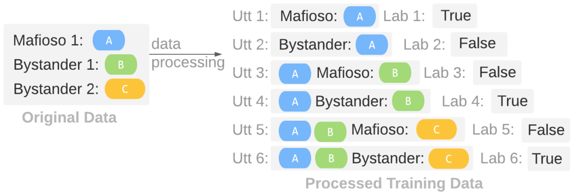


Figure 3: Data processing for fine-tuning BERT. The original data is shown on the left-hand side, while the right-hand side shows the processed data containing two versions of each utterance, one assuming that the target player is a mafioso and one assuming that they are a bystander, with the prior conversation context preceding each and labels corresponding to whether the assumed role matches the actual role of the player.

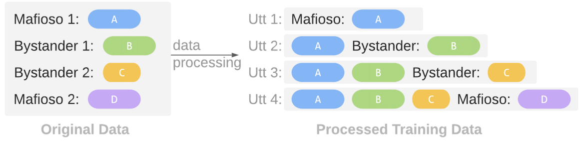


Figure 4: Data processing for fine-tuning GPT-2. The original data is shown on the left-hand side, while the right-hand side shows the processed data containing a version of the corresponding utterance with the prior conversation context preceding.

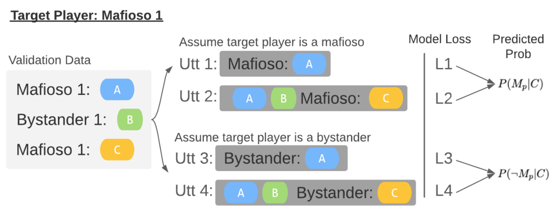


Figure 5: Prediction pipeline for GPT-2. Similar to the pipeline used to produce the training utterances, for prediction, there are now two versions of each, one assuming that the target player is a mafioso and one assuming that they are a bystander. The losses for each utterance of the target player are summed together in order to calculate the mafia and bystander probabilities as described in Equation 3.

introduce a hyperparameter $\alpha$[3] to adjust the relative influence of the prior and prediction:

$$P(M_p|C) = \frac{P(M_p)}{P(M_p) + P(\neg M_p)\left(\frac{P(\neg M_p|C;\theta_S)}{P(M_p|C;\theta_S)}\right)^{\alpha}} \tag{2}$$

### 4.2.2 Utterance Generation

To classify player $p$ using the auxiliary task of utterance generation, we fine-tune GPT-2 to generate utterance $C_i$ conditioned on prior utterances $C_{<i}$ and the role $S_i$ of the speaker that produced $C_i$. From Bayes' rule, we have $P(M_p|C) \propto P(M_p)P(C|M_p)$. To estimate $P(C|M_p)$, let $C_p$ include all $C_i$ for $i \in I_p$. We make the simplifying assumption that $P(C|M_p) \propto P(C_p|M_p)$, which assumes that the utterances made by players other than $p$ are independent of the role of player $p$. Then, if $M_p$ is true, $S_i$ is true for all $i \in I_p$, and so,

$$P(C_p|M_p;\theta_C) = \prod_{i \in I_p} P(C_i|C_{<i}, S_i;\theta_C).$$

Again, we introduce a hyperparameter $\alpha$[4] that adjusts the relative influence of the prior and prediction:

$$P(M_p|C) = \frac{P(M_p)}{P(M_p) + P(\neg M_p)\left(\frac{P(C_p|\neg M_p;\theta_C)}{P(C_p|M_p;\theta_C)}\right)^{\alpha}} \tag{3}$$

## 4.3 Data Processing

To train models for utterance classification (using BERT) and utterance generation (using GPT-2), we perform data processing procedures on the games' original dataset to create input representations $r(C, p)$ for each player $p$ and obtain our training datasets as shown in Figures 3 and 4. The left side of each figure shows a snippet of a game's data, where "Mafioso" and "Bystander" denote the true roles of the players. The utterances to the right of each figure are training data points used for fine-tuning the BERT and GPT-2 models. Structuring the data in this way provides both the prior context of utterances and the current utterance that happened within this context. This not only gives us the information that we seek for Questions 1 and 2 but also provides us with more training examples, as we only have 44 games in total. Moreover, this mimics the real game scenario from the bystander view in that they can only confirm their own role, but no one else's, which is the appropriate setting for us in which to detect deception.

Figure 5 shows the pipeline for using the GPT-2 model to predict players' roles. Let us assume that the target player for whom we want to predict their role is Mafioso 1. From the original game log on the left, we first perform the data processing scheme in Figure 4 twice, assuming that the target

[3] $\alpha$ was set to 1e6 for our experiments.
[4] $\alpha$ was set to 4.1 for our experiments.

player is a mafioso (top) and a bystander (bottom). Using our trained GPT-2 model, we then obtain a loss for each utterance denoted by L1 through L4. Summing all the losses for each role, as they denote log probabilities, we calculate $P(M_p|C)$ and $P(\neg M_p|C)$ via Equation 3. The target player's role as predicted by the model is finally given by comparing the two probabilities. A similar process is used to calculate $P(M_p|C)$ and $P(\neg M_p|C)$ for the BERT model.

# 5 Experiments

We train three fine-tuned models on the corpus of Mafia game records and compare their performance to a random baseline. The specifications for the baseline and models can be found below, and the results are shown in Table 1.

## 5.1 Random Baseline

This random classifier classifies each player as a mafioso or a bystander with probabilities equal to the prior distribution of each class. This serves as a baseline to be compared to for all other methods. In the game setting, this mimics a bystander player with only public information of how many mafia and bystanders are in the game.

## 5.2 Standard Classification

We initialize the model by loading a pre-trained BERT Base model (12 layers, 768 hidden dimension size, 12 attention heads). We train with a maximum sequence length of 256, which is sufficient for our post-processed dataset, setting the batch size to 16, the learning rate to 1e-5, and the maximum number of epochs to 25.

## 5.3 Utterance Classification

We initialize the model by loading a pre-trained BERT Base model (12 layers, 768 hidden dimension size, 12 attention heads). We train with a maximum sequence length of 512, which is sufficient for our post-processed dataset, setting the batch size to 5, the learning rate to 5e-5, and the maximum number of epochs to 25.

## 5.4 Utterance Generation

We initialize the model by loading a pre-trained 12-layer GPT-2 model with an embedding size of 768. For the dataset, we set the maximum length of each sentence to be 512, which is sufficient for our dataset after post-processing. During training, we set the batch size to be 5 and the learning rate to be 1e-5. We train the model for a maximum of 100 epochs.

## 5.5 Metrics

These approaches each estimate a probability $P(M_p|C)$ that a player $p$ is a mafioso given the full record of game texts $C$. In Mafia, bystanders do not declare who is and is not a mafioso, but instead vote each day to eliminate one of the players. Because the act of voting involves choosing one player among them all, a natural metric for evaluating the usefulness of a model is to order all players $p$ from greatest to least $P(M_p|C)$, their probability of being a mafioso under the model, and then to compute the average rank of the true mafia members. Therefore, the first metric in Table 1 is the average ranking of all mafia members when each player is ranked by $P(M_p|C)$ across the entire validation set composed of 5 games. It is also natural to consider player ranking within a single game, so we calculate the average ranking of mafia members within each game as a second metric. Smaller average ranking for mafia members means that the model is able to assign mafia players a high $P(M_p|C)$ relative to bystanders, which is desired.

In addition, we evaluate the accuracy of the classifiers and the precision and recall for each class.

## 5.6 Results and Analysis

We trained all models on 39 training games and evaluated on the remaining 5 validation games. The evaluation results are shown in Table 1. We have a total of 49 players in the validation games, but only considered the 39 players who had spoken at least one utterance throughout the game when calculating the metrics. As a rule, all other players (ie. those with no utterances) are given $P(M_p|C) = 0$.

First, we see that it is possible to achieve an average rank that is smaller than the random baseline, which demonstrates that there is information in the dialog about the roles of players, despite the fact that mafia members seek to hide their role while conversing. However, standard classification is comparable to random. Next, we observe that both models using auxiliary tasks outperform the standard classifier in rank-based metrics, which demonstrates that the auxiliary tasks provide useful inductive bias for the mafia classification task. Additionally, the accuracy is similar for all approaches, including random classification, which indicates

| | Avg Rank | Avg Rank/Game | Accuracy | Maf Prec | Maf Rec | Bys Prec | Bys Rec |
|---|---|---|---|---|---|---|---|
| **Random** | 19.0 | 3.4 | 0.62 | 0.26 | 0.26 | 0.74 | 0.74 |
| **Std Class** | 20.9 | 3.6 | **0.69** | 0.33 | 0.20 | 0.75 | **0.86** |
| **Utt Class** | 16.6 | 2.9 | 0.67 | **0.40** | 0.60 | 0.83 | 0.69 |
| **Utt Gen** | **11.4** | **2.1** | 0.64 | **0.40** | **0.80** | **0.89** | 0.59 |

Table 1: Experiment results on the validation set for random baseline (**Random**), standard classification (**Std Class**), utterance classification (**Utt Class**), and utterance generation (**Utt Gen**) approaches. Utterance generation outperforms all other methods in terms of average ranking overall and per game while also maintaining high accuracy, recall, and precision.

that there is not enough information in the text of a Mafia game for these models to determine players' roles reliably. If the goal of the game were to guess the role of each player individually, then always guessing bystander (ie. the majority class) would be the best strategy. However, since the goal for the bystanders is to vote to eliminate a mafia member each round, the utterance generation approach, which achieves the lowest average mafia ranking and average mafia ranking per game, is the most favorable.

Note that the precision for the mafia is much lower than that of the bystanders for all models. This is due to the usual lack of information available to predict that any player is a mafioso, which makes finding the mafia a much harder task than finding bystanders.

## 6 Discussion

The decoding ability of the GPT-2 model provides us a more straightforward way to understand what the model has learned. Given a prompt sentence, we can use our fine-tuned GPT-2 model to generate what a mafioso and a bystander would say. A few examples are shown in Table 2. From these examples, we inspect the following features that the model might be capturing to distinguish between mafia and bystanders: Feature 1: Referring to other players. Feature 2: Expressing confusion. Feature 3: Referring to others for elimination purposes. Feature 4: Asking for suggestions on who to eliminate.

To confirm that our fine-tuned GPT-2 model captures some of the above features, we hand-label these features on 5 training games and 1 validation game, obtain each players' feature vectors, and see whether there exists a correlation between the model's predicted $P(M_p|C)$ for validation players and the similarity of their feature vectors compared to the training set mafioso and bystander players.

| Prompt | Generated Utterance |
|---|---|
| lets kill P1. | **M:** sorry P1 :(<br>**B:** hello all |
| who thinks P3 is Mafia? | **M:** No i'm a bystander<br>**B:** No idea |
| That sounds suspicious... | **M:** P6 is mafia<br>**B:** Why yall want to eliminate me? |
| hi team. Hello!. Hi. | **M:** Who is the mob person?<br>**B:** hello |

Table 2: Utterances generated by our GPT-2 model given different prompts. **M** and **B** are shorthand for Mafioso and Bystander respectively, and P1, P3, and P6 denote the names of other players in the game.

| | Feat 1 | Feat 2 | Feat 3 | Feat 4 |
|---|---|---|---|---|
| **Mafioso** | 2.00 | 0.00 | 1.30 | 0.40 |
| **Bystander** | 1.06 | 0.27 | 0.65 | 0.10 |

Table 3: The average count per role for each of four hand-labeled features (number of references to other players, level of confusion, number of references to other players for elimination, and number of requests for who to eliminate) as identified by our GPT-2 model on 5 training games.

These feature vectors are shown in Table 3, where each entry denotes the average number of features per player of each role. As an example, for the first column, each mafioso player says 2 utterances having Feature 1 throughout the game on average, while each bystander player says 1.06 utterances having Feature 1 on average. We define the first row as a vector $v_1$ and the second row as $v_2$ for future references.

Table 4 shows the hand-labeled feature vectors for all 10 players in a validation game (first 4 columns, F1 to F4) ranked by the model's predicted $P(M_p|C)$. We define a metric function $D(u) = \|u - v_1\|^2 - \|u - v_2\|^2$ for a validation

| | F1 | F2 | F3 | F4 | D(u) | Pred | Truth |
|---|---|---|---|---|---|---|---|
| **P0** | 4 | 0 | 2 | 0 | -5.9 | 0.98 | B |
| **P1** | 2 | 0 | 2 | 0 | -2.1 | 0.93 | M |
| **P2** | 5 | 0 | 5 | 0 | -11.7 | 0.78 | M |
| **P3** | 2 | 0 | 2 | 0 | -2.1 | 0.63 | B |
| **P4** | 4 | 2 | 1 | 1 | -4.1 | 0.47 | B |
| **P5** | 3 | 0 | 2 | 0 | -4.0 | 0.43 | B |
| **P6** | 0 | 0 | 0 | 0 | 4.2 | 0.42 | B |
| **P7** | 1 | 0 | 1 | 0 | 1.0 | 0.40 | B |
| **P8** | 0 | 0 | 0 | 0 | 4.2 | 0.00 | B |
| **P9** | 0 | 0 | 0 | 0 | 4.2 | 0.00 | B |

Table 4: Features of each player (P0 to P9) in a validation game. For each row, F1 to F4 give the feature vector $u$ for the respective player. $D(u)$ gives the similarity of $u$ compared to the training feature vectors $v_1$ and $v_2$. Players are sorted by $Pred$, the probability $P(M_p|C)$ given by our GPT-2 model, and $Truth$ gives the true label for each player ($M$ for Mafioso, $B$ for Bystander). Since P8 and P9 have no utterances throughout the game, as per our heuristic, they are predicted to be bystanders with $P(M_p|C) = 0$.

player's feature vector $u$. The smaller $D(u)$ is, the closer $u$ is to $v_1$ than $v_2$, and hence the more mafia-like they are with respect to players in the training games. We can see that for players of higher rank, their $D(u)$ are negative with larger magnitudes. Referring to the true labels in the rightmost column ($M$ for Mafioso and $B$ for Bystander), the first row also explains how our model can fail to predict the true role of some players: even though this player is a bystander, they act more like the mafia than other bystanders according to these hand-labeled features because they are regularly referencing and accusing other players.

## 7 Conclusion & Future Work

We find that we are able to train models to differentiate players with different roles in the game of Mafia based only on their language use, as well as to identify features that may distinguish between these roles. We also noticed that the mafia were twice as likely to win the Mafia game than were the bystanders. These findings lead us to believe that the bystanders may benefit from being provided suggestions for whom to eliminate given our model's predictions and identified features. However, information that may aid bystanders may also aid mafia members in their deception.

How one uses language depends not only on the content they wish to convey, but also on the context within which they convey it, and speaker attributes such as conversational role contribute to such context. In this work, we leveraged an environment for which roles are explicitly labelled in order to make progress toward the task of deception detection, an essential task to protect users in our increasingly virtual world.

## References

Mohamed Abouelenien, Veronica Pérez-Rosas, Rada Mihalcea, and Mihai Burzo. 2014. Deception detection using a multimodal approach. In *Proceedings of the 16th International Conference on Multimodal Interaction*, pages 58–65.

Xiaoheng Bi and Tetsuro Tanaka. 2016. Human-side strategies in the Werewolf game against the stealth werewolf strategy. In *International Conference on Computers and Games*, pages 93–102. Springer.

Su Lin Blodgett and Brendan O'Connor. 2017. Racial disparity in natural language processing: A case study of social media African-American English. *arXiv preprint arXiv:1707.00061.*

Charles F Bond Jr and Bella M DePaulo. 2006. Accuracy of deception judgments. *Personality and Social Psychology Review*, 10(3):214–234.

Mark Braverman, Omid Etesami, and Elchanan Mossel. 2008. Mafia: A theoretical study of players and coalitions in a partial information environment. *The Annals of Applied Probability*, 18(3):825–846.

Judee K Burgoon, J Pete Blair, Tiantian Qin, and Jay F Nunamaker. 2003. Detecting deception through linguistic analysis. In *International Conference on Intelligence and Security Informatics*, pages 91–101. Springer.

Gokul Chittaranjan and Hayley Hung. 2010. Are you a werewolf? detecting deceptive roles and outcomes in a conversational role-playing game. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).*

Bob de Ruiter and George Kachergis. 2018. The Mafiascum dataset: A large text corpus for deception detection. *arXiv preprint arXiv:1811.07851.*

Sergey Demyanov, James Bailey, Kotagiri Ramamohanarao, and Christopher Leckie. 2015. Detection of deception in the Mafia party game. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, pages 335–342.

Douglas C Derrick, Thomas O Meservy, Jeffrey L Jenkins, Judee K Burgoon, and Jay F Nunamaker Jr. 2013. Detecting deceptive chat-based communication using typing behavior and message cues. *ACM Transactions on Management Information Systems (TMIS)*, 4(2):1–21.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Tommaso Fornaciari and Massimo Poesio. 2013. Automatic deception detection in Italian court cases. *Artificial Intelligence and Law*, 21(3):303–340.

Christie M Fuller, David P Biros, and Dursun Delen. 2011. An investigation of data and text mining methods for real world deception detection. *Expert Systems with Applications*, 38(7):8392–8398.

Michael Kearns, Stephen Judd, Jinsong Tan, and Jennifer Wortman. 2009. Behavioral experiments on biased voting in networks. In *Proceedings of the National Academy of Sciences 106.5*, pages 1347–1352.

Mike Lewis, Denis Yarats, Yann N. Dauphin, Devi Parikh, and Dhruv Batra. 2017. Deal or no deal? End-to-end learning for negotiation dialogues. *arXiv preprint arXiv:1706.05125*.

Piotr Migdał. 2010. A mathematical model of the Mafia game. *arXiv preprint arXiv:1009.1031*.

Rada Mihalcea, Verónica Pérez-Rosas, and Mihai Burzo. 2013. Automatic detection of deceit in verbal communication. In *Proceedings of the 15th ACM on International Conference on Multimodal Interaction*, pages 131–134.

Vlad Niculae, Srijan Kumar, Jordan Boyd-Graber, and Cristian Danescu-Niculescu-Mizil. 2015. Linguistic harbingers of betrayal: A case study on an online strategy game. *arXiv preprint arXiv:1506.04744*.

Jinie Pak and Lina Zhou. 2011. A social network based analysis of deceptive communication in online chat. In *Workshop on E-Business*, pages 55–65. Springer.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Gabriel Stanovsky, Noah A Smith, and Luke Zettlemoyer. 2019. Evaluating gender bias in machine translation. *arXiv preprint arXiv:1906.00591*.

Rachael Tatman. 2017. Gender and dialect bias in youtube's automatic captions. In *Proceedings of the First ACL Workshop on Ethics in Natural Language Processing*, pages 53–59.

Rachael Tatman and Conner Kasten. 2017. Effects of talker dialect, gender & race on accuracy of bing speech and youtube automatic captions. In *INTERSPEECH*, pages 934–938.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Lina Zhou and Yu-wei Sung. 2008. Cues to deception in online chinese groups. In *Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008)*, pages 146–146. IEEE.

## A Mafia Instructions

Below is a transcript of the instructions that were provided to participants before playing the Mafia game in our experiments:

"In this experiment, you will play a version of the party game "Mafia". You are going to play the game of Mafia (also known as Werewolf) with other participants. You are either part of the mafia (a mafioso) or a bystander. The mafia will know who is in the mafia, but the bystanders will not. There will always initially be more bystanders than mafia. There will be one or more mafia members. The goal of the mafia is to eliminate the bystanders one by one until the mafia are equal in number to them. The goal of the bystanders is to correctly guess the identity of the mafia and eliminate them all before the mafia win. There are two phases to this game, nighttime and daytime; at the end of each, a participant is eliminated from the game:

1. In the **nighttime** phase, only the mafia can converse and decide who they want to eliminate. Specifically, if you are a mafioso, you will talk in a chatroom, then use a drop-down menu to select who you want to remove. Mafia will have 1 minute to do this. If there is more than one mafioso and the mafia disagree about who to eliminate, one of the mafia's choices will be selected randomly. If you are a bystander, you will wait out this time, as you are sleeping during the night.

2. Everyone is awake during the **daytime** phase. The participant who was eliminated during the night will be announced: if you were eliminated, you will be sent to the end of the game and compensated. The remaining participants will converse (for 2 minutes and 30 seconds) and decide who to eliminate, where the goal of the bystanders is to eliminate a member of the mafia, and the goal of the mafia is to eliminate a bystander. By the end of this time, everyone needs to select a name from the drop-down menu. (If there are multiple mafia, the mafia will be reminded of each others' names in separate text on this page.) The participant with the most votes will be eliminated, except in the case of a tie, in which a randomly-selected vote will be eliminated. The eliminated participant and their identity (bystander or mafia) will be announced, and that participant will be sent to the end of the game and compensated.

The game will continue, alternating between nighttime and daytime, until either all of the mafia are removed (*bystanders win!*) or there are equal numbers of mafia and bystanders (*mafia win!*)"