www.nature.com/scientificreports

scientific reports

Check for updates

OPEN

Finding deceivers in social context with large language models and how to find them: the case of the Mafia game

Byunghwa Yoo[1] & Kyung-Joong Kim[1,2]✉

Lies are ubiquitous and often happen in social interactions. However, socially conducted deceptions make it hard to get data since people are unlikely to self-report their intentional deception behaviors, especially malicious ones. Social deduction games, a type of social game where deception is a key gameplay mechanic, can be a good alternative to studying social deceptions. Hence, we utilized large language models' (LLMs) high performance in solving complex scenarios that require reasoning and prompt engineering to detect deceivers in the game of Mafia given only partial information and found such an approach acquired better accuracy than previous BERT-based methods in human data and even surpassed human accuracy. Furthermore, we conducted extensive experiments and analyses to find out the strategies behind LLM's reasoning process so that humans could understand the gist of LLM's strategy.

A widely remarked human deception prevalence study[1] shows that people usually tell lies twice a day, meaning that lies are common in human social interactions. Unfortunately, although lies are such a common phenomenon in human social interactions, not all lies are harmless. Some lies are intentionally made to harass or exploit others by giving them false information.

Interestingly, state-of-the-art artificial intelligence (AI) models like large language models (LLMs) can tell false information that feels natural to humans[2–4]. Furthermore, they are shown to be able to detect human lies in non-social monologues[5]. Here, we can come up with an idea: Can LLMs detect intentional and malicious lies even in the social interaction of humans, i.e., in dialogues?

The problem with finding intentional lies in social contexts is that it is hard to get such data since nobody would self-report their intentionally malicious deceiving behaviors. Thus, smaller models of real social interactions—where the malicious, intentionally lying people are pre-defined—can be utilized to study social deception problems. Social deduction games are exact matches for such models; they are based on social interaction between humans and include hidden roles or relationships in the game, making deception a critical factor in the gameplay. Hence, social deduction games have been a popular subject in social network theories[6,7]. Not only social network theory studies but also AI studies[8,9] focus on social deduction games to study deception behaviors, but their researches lack either linguistic conversation in the environment or information partiality, which are all critical facts in social interactions.

On our experiments, LLMs were given real human data of playing the game Mafia. LLMs, as an observer of the games, were asked to predict who the mafia are given partial recording of the games. As said in the previous paragraph, the act of finding mafioso is equivalent to finding deceiving actors in social contexts. Since LLMs have shown state-of-the-art performances on very complex tasks that require reasoning or planning[10–12], we expect such ability of LLMs would make usage in predicting deceivers in social contexts.

Our work is the first study to utilize state-of-the-art LLMs' high performance in solving complex scenarios that needs reasoning in the field of finding deceivers in the social context of real human data, at least up to our knowledge. We conduct an extensive analysis of how LLMs decide malicious deceivers in the social context given only partial conversational information in the game of Mafia, a popular type of social deduction game. Furthermore, we exploit the explainability of LLMs to give humans a comprehension of how to find deceivers in social contexts.

[1]AI Graduate School, Gwangju Institute of Science and Technology, Gwangju 61005, Republic of Korea. [2]School of Integrated Technology, Gwangju Institute of Science and Technology, Gwangju 61005, Republic of Korea. ✉email: kjkim@gist.ac.kr

Background and related works

Artificial intelligence-based approaches for social deduction games

Social deduction games have been a research topic for various academic fields. Earlier studies included biased voting in social network[6], how culture affected cues of deception using Chinese social deduction game[13], social network-based deceiver detection in computer-mediated communication using Mafia game website epicmafia.com[7], and mathematical model developments for social deduction games[14,15].

While earlier works tried to use social deduction games in fields of social network systems and mathematics, more recent studies tried to develop AI models regarding social deduction games. Nakamura et al.[16] built a psychology-based AI model that could play the social deduction game of Werewolf. Serrino et al.[8] developed a multi-agent model that could surpass human abilities in the social deduction game The Resistance: Avalon. Similarly, Chuchro[17] tried to develop an assassin agent in the same game, The Resistance: Avalon. However, although these studies were able to get decent performances, their settings lacked linguistic information—making their environment far apart from real-life social interactions.

Thus, considering most real-life social interactions and social deduction games include conversations, it is critical to include linguistic information in the environment. One of the first research to consider linguistic information was de Ruiter et al.[18], yet this study utilized explicit features such as a number of specific words, not the semantic meaning of conversations. However, incorporating linguistic information into the environment was known to be a difficult problem, thus, several studies utilize modern NLP approaches such as fine-tuning PLMs. Lai et al.[19] viewed social deduction games as persuasive social interactions and tried to develop a multi-modal model that could decide what persuasive strategy was used in the game, where their model was built based on fine-tuning Ppe-trained language models (PLMs) such as BERT[20] and RoBERTa[21]. On the other hand, Ibraheem et al.[9] focused on deceptions in social deduction games. They fine-tuned BERT and GPT-2[22] to find deceivers given the full record of Mafia game dialogues. We also aimed to find deceivers in the Mafia, but we conducted experiments with the partial environment to further enhance real-life usability.

Large language models

LLMs are large models, usually transformer[23] architecture trained with large corpora. Although there was no strict definition of how big the model size or corpora should be for some models to be LLMs, we considered the term LLM in this study to describe generative models with larger size than GPT-2, including GPT-3[24], GPT-4[25], and LLaMA[26].

Thanks to large size and generalized performances, LLMs showed remarkable results in various fields even without fine-tuning with additional data with the help of prompt engineering, the act of giving appropriate inputs to LLMs. Furthermore, to improve performances of prompt engineering, techniques like the chain of thoughts[27] (CoT) or the tree of thoughts[28] were proposed for few-shot prompts, and zero-shot CoT[29] was used for zero-shot cases. Plus, such mechanisms also performed as explainable AIs, since using these lets LLMs give their analyses behind their decisions when giving the output.

Plus, LLMs have shown very high performances in complex scenarios requiring reasoning or planning, which was unsuitable for smaller language models. LLMs have shown high performance in various complex tasks requiring planning or reasoning, including legal reasoning[10], video games[11,12], etc.

Methods

The Mafia game

The Mafia game, or the Werewolf game, is a popular social deduction game developed by Dimitry Davidoff in 1986. The mafia game has two teams—the uninformed majority team called bystanders (villagers) and the informed minority team called the mafia (werewolves). The bystanders' goal is to remove all mafia hiding among bystanders while bystanders get no information on who the mafia is. On the other hand, the mafia's goal is to remove bystanders until the number of remaining bystanders and the mafia becomes the same before all mafia are executed by bystanders. Plus, the mafia members are informed about who the mafia is before the start of the game. The game continues with two alternating phases—daytime and nighttime. During the daytime, all bystanders and the mafia vote for one person to be executed. During the nighttime, the mafia decides one person to kill, and the nighttime conversation cannot be heard by bystanders.

Consider a simple mafia game walkthrough with six members—A, B, C, D, E, and F—where A and B are mafia, C, D, E, and F are bystanders. The game starts with the nighttime phase. On 1st nighttime, A and B decide to kill C. Then on 1st daytime, C is revealed as a victim. All A, B, D, E, and F have open conversations and open voting. By the voting, A was decided to be executed. On the following 2nd nighttime, only remaining mafioso B kills D. Then on the 2nd daytime, remaining B, E, and F have an open vote, and E was decided to be executed. Then the mafia team won the game since the number of remaining mafia and the remaining bystanders were the same 1 person.

In our experiment, we used the same rules in the walkthrough, where no additional roles besides the mafia and bystanders were used.

Dataset

We used the Mafia game dataset from Ibraheem et al.[9], which collected dozens of game-playing dialogues of the Mafia game, where the games followed the rules described in The Mafia Game subsection. The data was collected via Amazon Mechanical Turk, having a total of 460 English-speaking participants.

Table 1 shows the number of Mafia games included in the training, test, and invalid datasets. Data being collected online, some games were not valid to be used for our study. The criterion for the invalid data was whether there was no victim of the mafia, either it was due to technical issues or mafia players were not properly

Train	Test	Invalid
23	15	6

Table 1. Number of games for training data, test data, and invalid data.

	Train	Test
Maximum participants	12	12
Mean participants	9.58	9.73
Minimum participants	7	7
Maximum game length (days)	5	4
Mean game length (days)	3.21	3.07
Minimum game length (days)	1	2
The ratio of 3 days or shorter games	0.63	0.73
Mean exchanges per game	15.67	15.60
Mean sentences used per game	31.45	33.6
Mean words used per game	214.0	211.73

Table 2. Number of participants and lengths of games in training data and test data.

Length	Train	Test
Up to day 2	23	15
Up to day 3	21	12
Up to day 4	10	4
Up to day 5	2	0

Table 3. Number of samples in train and test dataset according to length.

participating in the game. 6 games of Mafia had no victims, and it was decided they were invalid samples. Although our approach was focused on state-of-the-art LLMs and prompt engineering, BERT-based baselines utilized fine-tuning, meaning that training data was needed for performance analysis. For the training dataset for baselines, we used 23 games of Mafia, and for the test, we used 15 games of Mafia. Training and test datasets were randomly split, except for invalid ones.

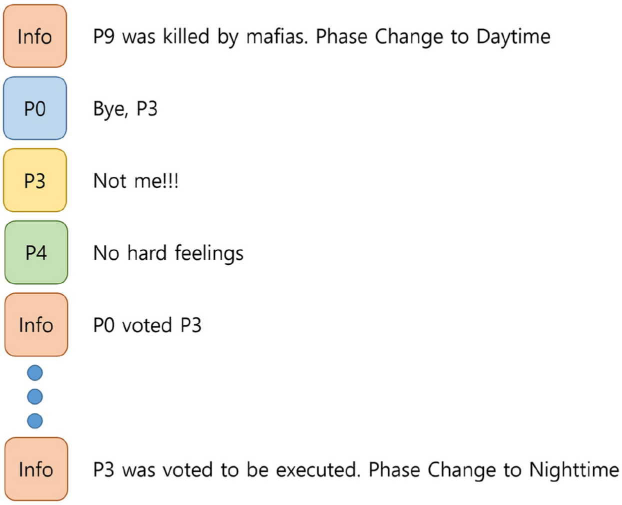
Table 2 shows the number of people who participated in the games and the length of the games in the training and test dataset. Both training and test data had a maximum of 12 people and a minimum of 7 people participating for each game in the data and had similar mean participants of 9.58 and 9.73 people. For the lengths of the games, game lengths were not fixed. Both training and test data had a mean length of slightly more than 3 days per game, and approximately 60–70% of games lasted 3 days or shorter. For linguistic information on the data, in both train data and test data, there were around 15.6 exchanges (number of utterances), and around 30 sentences were used per game, and 200 words were used per game.

Unlike considering the full course of the game, i.e., from the start of the game to the end, we considered partial data of the Mafia game, i.e., input that only contains information up to a specific point of the game, not the end. To test our baselines on partial information, game data were split into units of days. For each sample of data, we let each sample include from the start of the game to the end of a specific day, at the point of execution victim by voting was published. The reason why we set the data format like this was to make machines have similar inputs with human players of the Mafia game; human players had information only on what was previously said before they voted. Table 3 shows how much partial game dialogue samples were used for training and testing.

Figure 1 shows an example explaining the format of the data used for training and testing. As shown in the figure, each people were anonymized as P0, P1,..., and P11 to exclude the effect of names in training and inferencing. Note that the order of name anonymization was changed in every sample, i.e., the same person referenced as P0 in one sample might appear as P7 or P8 in other samples. Furthermore, non-conversational information such as someone being killed by the mafia or someone voting for another, was included as non-participating player "information" stating such necessary information. Plus, no identities of the victims, including both mafia victims and execution victims, were included in the data, since if any victim was revealed as a mafioso then it would be ensured to predict at least one mafia correctly for any models in any case.

Baselines and metrics

We used 5 types of baselines for our experiments: GPT-4, GPT-3.5-turbo (ChatGPT), BERT-Multilabel, BERT-Utterance, and Random. GPT-4 (estimated 1.76 trillion parameters) and GPT-3.5-turbo (175 billion parameters), both widely used commercial LLMs trained with an extensive size of corpora by OpenAI, were used as baselines to show how LLMs' high performance on reasoning requiring complex tasks affects deception detection in the



Info P9 was killed by mafias. Phase Change to Daytime

P0 Bye, P3

P3 Not me!!!

P4 No hard feelings

Info P0 voted P3

Info P3 was voted to be executed. Phase Change to Nighttime

Fig. 1. Example of an input sample showing the start of 1 day to its end.

social context. On the other hand, BERT-Multilabel, and BERT-Utterance, were BERT-based classifier models and used as baselines to show relative performances against commercial LLMs.

For *GPT-4* and *GPT-3.5-turbo*, we used prompt engineering and zero-shot CoT. Figure 2 shows the prompt we used in our setting. As shown in Fig. 2, all rules and necessary information were given in *SYSTEM*, and game dialogues were given in *USER*. The phrase “step-by-step” was included in rule 6 and the LLMs were asked to write the reason in rule 7 under the *SYSTEM* to invoke the zero-shot CoT process to enhance performance. Temperatures were set to 0 to exclude the randomness of LLMs.

BERT-Multilabel was a naïve BERT-based model to decide who the mafia were. Since there were at most 12 participants, the BERT model was fine-tuned with training data to perform multi-label classification for 12 classes. Since all games had two mafias, the model was trained to predict the two most likely mafia labels given the input partial dialogue. For testing, two classes with the highest probabilities were regarded as the model’s prediction for two mafias.

BERT-Utterance is a BERT-based model originally introduced to detect mafia via binary classification in Ibraheem et al.[9] Since the original model was binary classification given the full record of the game, we re-trained and modified the model so that it can be used with partial game dialogues and be tested with the same metrics used in the other baselines. Figure 3 shows the modified input for determining if P0 was the mafioso given the modified input. As shown in the figure, the player to be checked was tagged with the word “mafioso” for every single utterance the person said, i.e., to check P3, the word “mafioso” would be tagged to P3. Given such a form of input, the BERT-utterance model returned the probability of the player being a mafioso. For testing, every single player was tested per data sample, and the two players with the highest probabilities were regarded as the model’s prediction for the two mafias. Note that people with no utterance cannot be predicted using this model—they were strictly given the probability 0 of being mafias.

Finally, for the random baseline, the random model was asked to randomly select any 2 different participants among 12 candidates, since 12 is the maximum number of classes used in BERT-multiclass. The random baseline used uniform random variables and were tested 100 times.

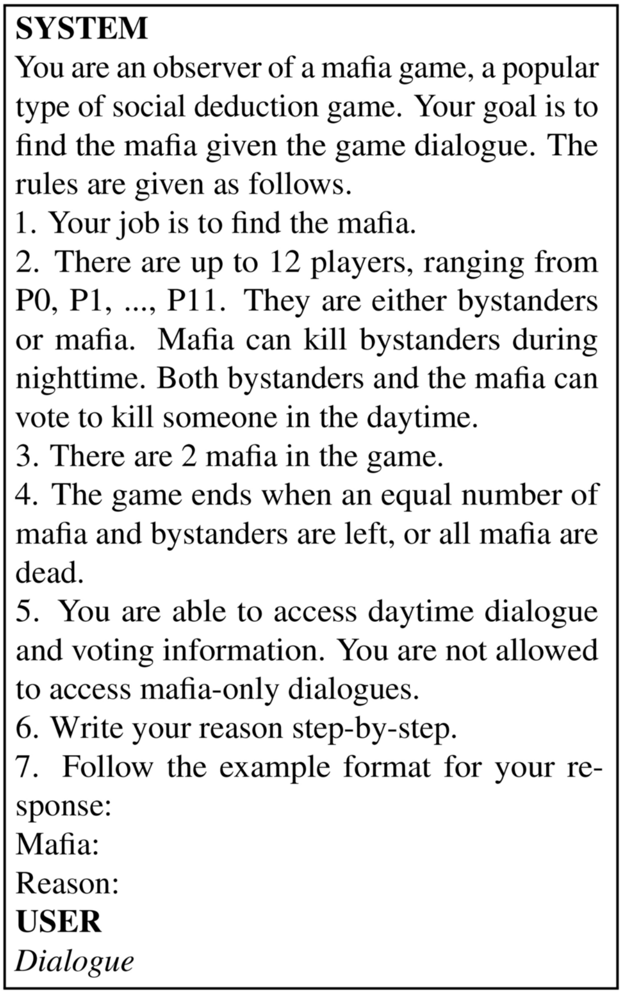
For both *BERT-Multilabel* and *BERT-Utterance* models, they were fine-tuned from bert-base-uncased[20], using AdamW optimizer[30] with learning rate 5e−5, weight decay of 0.01, and the batch size of 8. The models were trained for at most 20 epochs with the loss function of cross entropy, and the epoch with the best validation loss was used for inference.

To evaluate the performance of the baselines, we used two metrics, single-match accuracy and exact-match accuracy. *Single-Match Accuracy* was defined as the ratio of samples that models predicted at least one mafioso correctly. On the other hand, *Exact-Match Accuracy* was defined as the ratio of samples that models predicted both two mafias correctly.

Results

Tables 4 and 5 show the single match accuracy and exact match accuracy of four baseline models. GPT-4 and GPT-3.5-turbo used temperatures of 0 so that their answers were definite, and for BERT-based models, they were trained 5 times each; mean accuracy and standard deviations are shown in Tables 4 and 5.

In both single-match accuracy and exact-match accuracy metrics, GPT-4 showed the highest accuracy compared to the other baselines. In single-match accuracy, GPT-4 showed 80.65% accuracy, which was more than 24% higher than the BERT-Utterance model’s second-best accuracy of 56.13%. On the other hand, GPT-3.5-turbo showed the lowest accuracy of 41.94%, which was slightly lower than BERT-Multilabel’s 45.16% and BERT-Utterance’s 56.13%. In exact-match accuracy, GPT-4 showed 19.35% accuracy, which was significant compared to the other models. GPT-3.5-turbo and BERT-Multilabel were not able to predict correctly at all, getting 0% accuracy.



SYSTEM

You are an observer of a mafia game, a popular type of social deduction game. Your goal is to find the mafia given the game dialogue. The rules are given as follows.

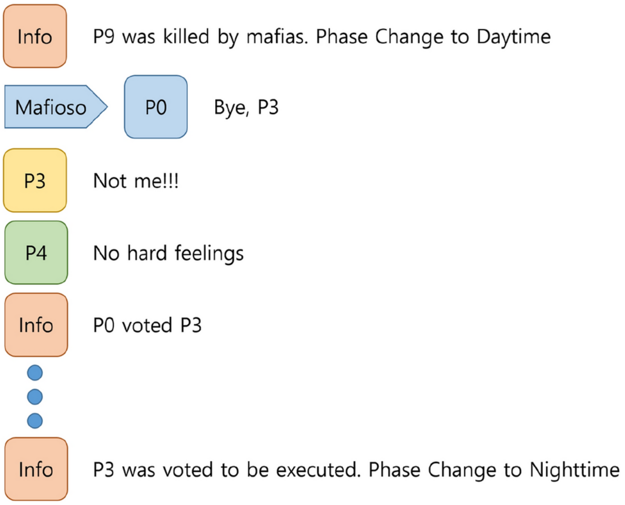
1. Your job is to find the mafia.
2. There are up to 12 players, ranging from P0, P1, ..., P11. They are either bystanders or mafia. Mafia can kill bystanders during nighttime. Both bystanders and the mafia can vote to kill someone in the daytime.
3. There are 2 mafia in the game.
4. The game ends when an equal number of mafia and bystanders are left, or all mafia are dead.
5. You are able to access daytime dialogue and voting information. You are not allowed to access mafia-only dialogues.
6. Write your reason step-by-step.
7. Follow the example format for your response:

Mafia:
Reason:

USER

Dialogue

Fig. 2. Prompt applied to GPT-4 and GPT-3.5-turbo models. Game dialogues were included in *Dialogue* part under *USER* section.



P9 was killed by mafias. Phase Change to Daytime

Bye, P3

Not me!!!

No hard feelings

P0 voted P3

P3 was voted to be executed. Phase Change to Nighttime

Fig. 3. Example of an input sample for determining if P0 is the mafioso in the game using BERT-Utterance model.

Info length	GPT-4 (%)	GPT-3.5-turbo (%)	BERT-multilabel (%)	BERT-utterance (%)	Random (%)
Up to day 2	80.00	57.14	33.33 ± 0.00	61.33 ± 7.77	31.73 ± 4.07
Up to day 3	75.00	33.33	58.33 ± 0.00	55.00 ± 4.08	32.67 ± 5.79
Up to day 4	100.00	20.00	50.00 ± 0.00	40.00 ± 20.00	38.75 ± 4.02
Total	80.65	41.94	45.16 ± 0.00	56.13 ± 2.58	33.00 ± 5.31

Table 4. Single match accuracy of our baselines on the test dataset. For GPT-4 and GPT-3.5-turbo, temperatures were set to 0 to make the models deterministic. For BERT-based models, the models were trained 5 times, and the mean and the standard deviation were recorded. For the Random baseline, it was tested 100 times. Significant values are in bold.

Info length	GPT-4 (%)	GPT-3.5-turbo (%)	BERT-multilabel (%)	BERT-utterance (%)	Random (%)
Up to day 2	13.33	0.00	0.00 ± 0.00	16.00 ± 5.33	1.53 ± 1.26
Up to day 3	**33.33**	0.00	0.00 ± 0.00	6.67 ± 3.33	1.91 ± 1.32
Up to day 4	0.00	0.00	0.00 ± 0.00	0.00 ± 0.00	2.00 ± 1.00
Total	**19.35**	0.00	0.00 ± 0.00	10.32 ± 3.76	1.74 ± 1.27

Table 5. Exact match accuracy of our baselines on the test dataset. For GPT-4 and GPT-3.5-turbo, temperatures were set to 0 to make the models deterministic. For BERT-based models, the models were trained 5 times, and the mean and the standard deviation were recorded. For the Random baseline, it was tested 100 times. Significant values are in bold.

Info length	GPT-4 (%)	Participants (%)
Up to day 2	33.33	33.33 ± 40.37
Up to day 3	**50.00**	32.87 ± 31.31
Up to day 4	**75.00**	14.58 ± 15.16
Total	**45.16**	30.73 ± 35.16

Table 6. Accuracy of data participants and GPT-4 in the test dataset. Significant values are in bold.

In both single-match and exact-match cases, GPT-3.5-turbo got the lowest mafia prediction performance, even though it was a commercial LLM trained on an extensive size of corpus and known to gain public syndrome for chatGPT due to its performances in various tasks. We suspected that this was since LLMs' performance is highly dependent on their size[24,25], i.e., the larger the model gets, the better the performance becomes. We believe our study was also one of such cases. However, GPT-4's performance and GPT-3.5-turbo's performances varied significantly, almost twice in single-match accuracy and GPT-3.5-turbo getting 0 exact-match accuracy, we believe there was a critical point of LLMs' parameter sizes and performance that enables better prediction ability in finding deceivers in the social context.

Performance compared to data collection participants

As stated in The Mafia Game subsection, the players voted for a person to be executed during the daytime phase, meaning bystanders should guess who the mafioso was and vote for that person to be executed. Therefore, we could retrieve the ratio of bystanders correctly voting for the mafioso by getting the number of total bystanders' votes and the number of bystanders who correctly voted for the mafia. When retrieving mafia prediction accuracy of data participants, i.e., human players in the data collection, we applied the Eq. (1) to each sample of the test dataset. Note that abstention votes were not included in the denominator.

$$Accuracy = \frac{\#\text{Bystanders voted to mafia}}{\#\text{Bystanders who voted}}. \tag{1}$$

However, the baselines were asked to get all mafia, while participants of the game could vote for only one person each day. Plus, the baselines consider the possibility of players already executed or murdered being mafias, while humans voted only for the surviving players. Hence, we added one more rule to equalize baseline conditions with human participants, *"Pick only one player most likely to be a mafioso. Exclude executed players."* to pre-existing rules in the GPT-4's SYSTEM shown in Fig. 2, so that GPT-4 could only predict one player most likely to be a mafioso, excluding the players already dead.

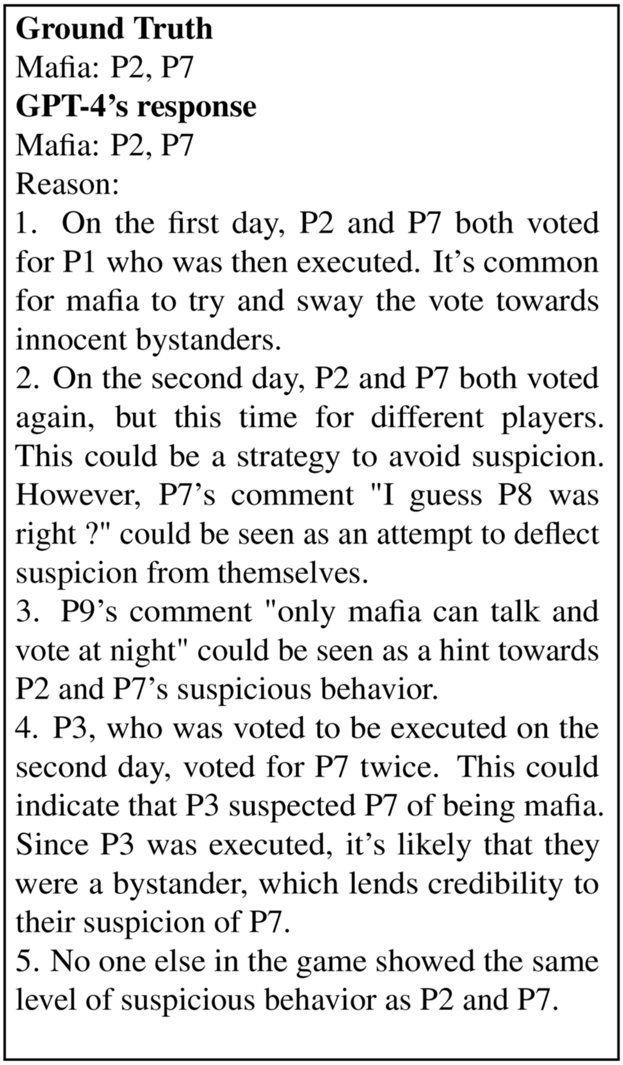
Table 6 shows the performance of GPT-4 versus the retrieved performance of human participants in predicting the mafia in-game. It was shown that GPT-4 showed a much higher mafia prediction performance of 45.16%, which was more than 1.5 times more accurate than human participants' 28.83%. Furthermore, GPT-4 always showed higher performance compared to human participants regardless of how much data was given as input, either in small fragments of game dialogues (up to day 2) or full-course dialogues (up to day 4 or later).

An interesting trend could be found when comparing the trend of prediction accuracy against how much information was given. For GPT-4, it was shown that the more information was given, the mafia prediction accuracy increased from 33.33% → 50.00% → 75.00%. On the other hand, human participants' mafia prediction accuracy seemed to decrease when more information was given, showing 33.33% → 32.87% → 14.58%. This could be perceived as that GPT-4 was good at detecting mafias' lies while human participant bystanders were played by mafias' lies in late-game situations.

The reasons behind LLM's decisions

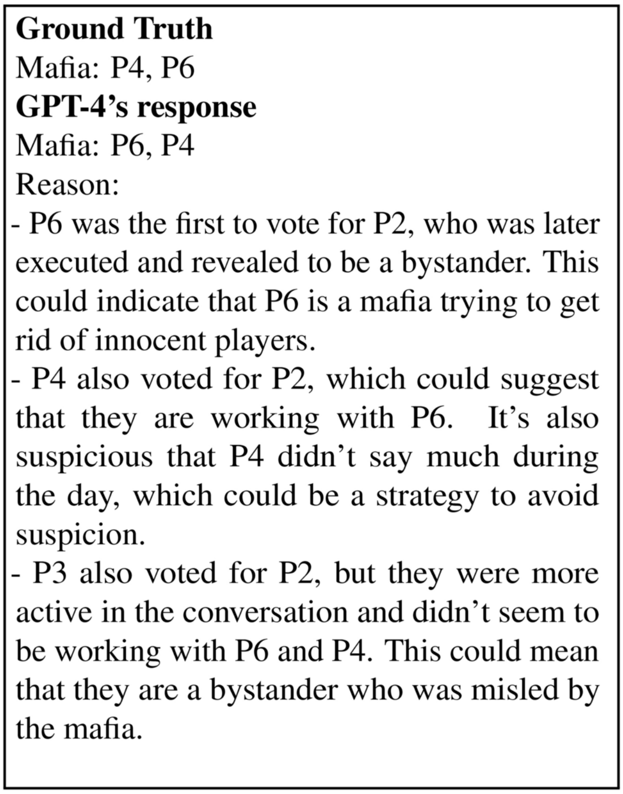
LLMs, by their nature, are text-to-text generative AI. Such characteristics of LLMs let them to be able to generate analyses of the reasons behind their decisions. Furthermore, to use CoT processes, printing the thinking process in detail was encouraged for LLMs. Our GPT-4 and PGT-3.5-turbo models utilized the zero-shot CoT process, also encouraging the GPT models to print out their reason analyses. We could exploit such processes to explainability for humans, although their reasoning logic might not be perfect.

Figures 4 and 5 show two examples of reasons for GPT-4's reasons for its decisions. Although for both cases the predictions were exactly correct, not all parts of their reasoning were logically correct. For the case of Fig. 4, GPT-4 was able to conclude who were the mafias very well with good reasoning. On reasons 1 and 2, GPT-4 suspected P2 and P7 as culprits since they voted for the same victim P1 on day 1, and it reinforced its suspicion as P2 and P7 voted for different people on day 2, hypothesizing the culprits were using suspicion deflection strategy. On reasons 3 and 4, GPT-4 utilized comments or voting information of other participants to decide the culprits. By combining all of the factors, GPT-4 was able to conclude that P2 and P7 were comparably most likely to be mafia.



Ground Truth
Mafia: P2, P7
GPT-4's response
Mafia: P2, P7
Reason:
1. On the first day, P2 and P7 both voted for P1 who was then executed. It's common for mafia to try and sway the vote towards innocent bystanders.
2. On the second day, P2 and P7 both voted again, but this time for different players. This could be a strategy to avoid suspicion. However, P7's comment "I guess P8 was right ?" could be seen as an attempt to deflect suspicion from themselves.
3. P9's comment "only mafia can talk and vote at night" could be seen as a hint towards P2 and P7's suspicious behavior.
4. P3, who was voted to be executed on the second day, voted for P7 twice. This could indicate that P3 suspected P7 of being mafia. Since P3 was executed, it's likely that they were a bystander, which lends credibility to their suspicion of P7.
5. No one else in the game showed the same level of suspicious behavior as P2 and P7.

Fig. 4. GPT-4's good reasoning example.



Ground Truth

Mafia: P4, P6

GPT-4's response

Mafia: P6, P4

Reason:

- P6 was the first to vote for P2, who was later executed and revealed to be a bystander. This could indicate that P6 is a mafia trying to get rid of innocent players.
- P4 also voted for P2, which could suggest that they are working with P6. It's also suspicious that P4 didn't say much during the day, which could be a strategy to avoid suspicion.
- P3 also voted for P2, but they were more active in the conversation and didn't seem to be working with P6 and P4. This could mean that they are a bystander who was misled by the mafia.

Fig. 5. GPT-4's reasoning example. The predictions were correct, however, the reasoning analyses were not completely sound.

However, in Fig. 5 case, GPT-4 was not able to get fully valid reasoning analyses. It was able to deduce P4, and P6 were mafia based on voting information and the fact both voted for the same P2. However, it considered P2, a voting execution victim, to be a bystander without any specific reason, meaning that GPT-4 did not perfectly understand the rules or made oversimplifications.

In conclusion, we could say the GPT-4 model was able to generate an analysis of reasons behind their decisions that keeps a logical gist that humans could perceive, but it did not guarantee concrete reasons. However, there is a concern that such analyses could be interpreted as a legitimate thinking process of LLMs. Yet, even if such analyses are more post-hoc analyses, considering that LLMs can generate meaningful post-hoc analyses[31], we could say such analyses still have the logical gist for humans to consider.

Ablation studies

Voting information

In the previous subsection, it was shown that voting information was critical in GPT-4's prediction process. Hence, it would be a good idea to check two things regarding how significant factor voting was to GPT-4. First, we removed voting information from the input, so that we could see how considerable voting was in the process. Second, we give only voting information to the GPT-4. Such a situation made linguistic information excluded as much as possible, being semantically similar to a non-language-based social deduction environment in Serrino et al.[8] We tested this case to see how critical other information outside voting was in the reasoning process.

Table 7 shows the results of information without voting and voting information only cases of GPT-4 compared to original GPT-4 results in both single-match and exact-match accuracy metrics. It was shown that removing either voting information or using only voting information proved to decrease accuracy by a significant amount. Overall, removing voting information proved to be better than using only voting information acquiring 64.52% and 9.68% accuracy for single-match and exact-match respectively, while using only voting information gained significantly low accuracy of 22.58% and 0.00%. This might be interpreted as that although voting was a key factor in GPT-4's prediction processes, non-voting conversations did play a significant role in interpreting the Mafia game.

Changing temperatures of LLMs

Figure 6 shows the mafia detection accuracy of GPT-4 models with different temperatures. We tested temperatures varying from 0 to 1, for each case testing 3 times except the deterministic case of temperature 0. Although it was known that increasing temperature increased the creativity of LLMs, so higher temperature may help or hinder performance according to what task it was dealing with, our case was shown to be unaffected by the temperature changes. Except for temperature 1.0 cases, all GPT-4 models with different temperatures showed

Info length	Original (%)	w/o vote (%)	Only vote (%)
Single-match accuracy			
Up to day 2	**80.00**	46.67	6.67
Up to day 3	75.00	**83.33**	33.33
Up to day 4	**100.00**	75.00	50.00
Total	**80.65**	64.52	22.58
Exact-match accuracy			
Up to day 2	**13.33**	6.67	0.00
Up to day 3	**33.33**	16.67	0.00
Up to day 4	0.00	0.00	0.00
Total	**19.35**	9.68	0.00

Table 7. Single-match and exact-match accuracy of original GPT-4, GPT-4 without voting information, and GPT-4 with only voting information. Temperatures were set to 0 to make models deterministic. Significant values are in bold.

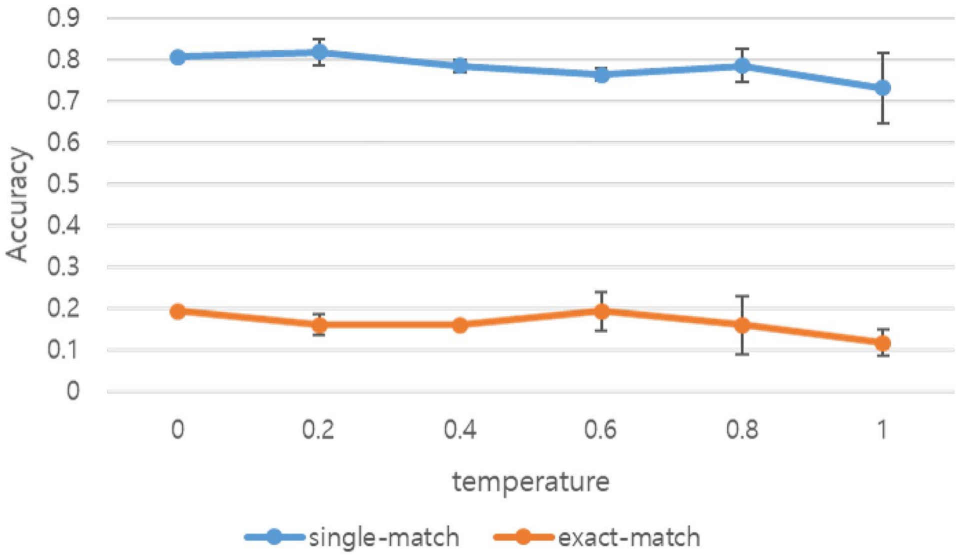


Fig. 6. Single-match and exact-match mafia detection accuracy of GPT-4 models differing temperatures. For all models except temperature 0, models were tested 3 times. Means and standard deviations are shown in the graph.

accuracy between 0.75 and 0.82 in single-match accuracy and between 0.16 and 0.2 in exact-match accuracy, which were no significant differences. Similarly, Renze et al.[32] reported that LLMs do not show meaningful performance differences in multiple-choice question-and-answer (MCQA) problems. Mafia detection could be considered a sort of MCQA—finding mafias given several players—it would be natural for GPT-4 not to be affected much by temperatures.

Limitations

Although our study showed the LLMs' power to predict deceivers in social contexts and a limited extent of explainability, there are several important limitations our study had. First, there was a data shortage problem. Since our experiment could use data from only 15 games, two big problems occurred. One was that our results for up to day 4 were limited by the very small number of data, which did not provide sufficient statistical power to draw definitive conclusions. Given the limited data, especially in the later stages of the game, these findings should be interpreted with caution. Another is that having too little data may not be sufficient to train BERT-based models; larger data may yield better prediction ability for these models. Second, the explainability of GPT-4 for social deception was rather limited. GPT-4 produced misleading analyses about their decisions and sometimes showed doubtful results if GPT-4 is truly getting the rules of the game completely correct. Furthermore, it was impossible to create a quantitative standard to measure whether GPT-4's explanations were valid, or how much were valid. Finally, there was a model diversity problem. Our experiment only included OpenAI's two commercial models as LLMs, yet there are many open-source LLMs such as LLaMA, which could be further fine-tuned, which means that there might be a chance for fine-tunable LLMs that could outperform GPT-4, but we were not able to test that.

Discussion

In this study, we aimed to find deceivers in human data via the commonsense reasoning power of LLMs. We showed that GPT-4 was able to find deceivers very well, achieving higher accuracy than BERT and GPT-3.5-turbo, and even humans. Plus, by using the generative nature of LLMs, we showed that GPT-4 could generate analyses on how it decided its answers to some extent, possibly giving limited insight to humans.

In the comparison of GPT-4 against human participants in Table 6, we can see that GPT-4's accuracy rises as more data is given while participants (bystanders) show lower accuracy. There could be several reasons we can consider for such results. First, unless mafia members are executed early in the game, the ratio of mafia among total surviving members goes up, probably leading to a more mafia-friendly public opinion environment. Meanwhile, GPT-4 was in the position of a pure observer, i.e., not inside the game, giving them more robustness against public opinion control. Second, although humans act based on memory, it is not that all participants keep track of every part of the whole game dialogues, and the longer the game, the harder for humans to remember the whole game. On the other hand, GPT-4 processes input data at once, meaning that all of the game dialogues in early parts and later parts could be processed well. Similarly, if we compare GPT-4 with BERT-based methods or GPT-3.5-turbo, GPT-3.5 gained less prediction accuracy given more information, and BERT-based models seem not affected much by the data amount in the single-match accuracy in Table 4. The factor behind this might be that these models, unlike GPT-4, were having problems handling large amounts of data in late-game dialogues, considering that GPT-4 has an extremely larger size compared to the others.

If we look into the qualitative analyses of the GPT-4's responses, they often generate misleading explanation analyses for their decisions or even wrong interpretations of the game's rules. Various factors might affect results. A factor that might affect the analysis performances is the stochastic property of GPTs. Considering that a high temperature of 1.0 led to a lower prediction score in Fig. 6, more strict restrictions to reduce stochastic property may help. Furthermore, the input about the rules of the game was likely to affect the reasoning analyses' validity, such as the term execution being directly considered bystander's death in Fig. 5, which implies that the term execution was not perfectly understood as a game context term by GPT-4. Yet, we were not able to find a better system prompt to solve this problem.

Although GPT-4's explanation performance was limited and often misleading, we believe that our study showed some level of explainability by GPT-4. We believe our work to be an early work for finding deceivers in a social context via LLM, and we believe our study implies that larger and better models that will come in the future will have better explainability.

For future works, we aim to develop our system in more complex scenarios, such as real-world human social interactions or social deduction games with more rules and larger participants. Furthermore, we would try fine-tuning open-source LLMs such as LLaMA if we are to gain more abundant data for social deduction games.

Data availability

For the Mafia Game dataset, we used a publicly open dataset first introduced in "Putting the Con in Context: Identifying Deceptive Actors in the Game of Mafia" (Ibraheem et al. 2022), which can be found at: https://github.com/omonida/mafia-dataset/tree/main.

Received: 19 June 2024; Accepted: 2 December 2024

Published online: 28 December 2024

References

1. DePaulo, B. M., Kashy, D. A., Kirkendol, S. E., Wyer, M. M. & Epstein, J. A. Lying in everyday life. *J. Pers. Soc. Psychol.* **70**, 979. https://doi.org/10.1037/0022-3514.70.5.979 (1996).
2. Azaria, A. & Mitchell, T. The internal state of an llm knows when its lying. Preprint at http://arxiv.org/abs/2304.13734 (2023).
3. Park, P. S., Goldstein, S., O'Gara, A., Chen, M. & Hendrycks, D. Ai deception: A survey of examples, risks, and potential solutions. Preprint at http://arxiv.org/abs/2308.14752 (2023).
4. Pacchiardi, L. et al. How to catch an ai liar: Lie detection in black-box llms by asking unrelated questions. Preprint at http://arxiv.org/abs/2309.15840 (2023).
5. Loconte, R., Russo, R., Capuozzo, P., Pietrini, P. & Sartori, G. Verbal lie detection using large language models. *Sci. Rep.* **13**, 22849. https://doi.org/10.1038/s41598-023-50214-0 (2023).
6. Kearns, M., Judd, S., Tan, J. & Wortman, J. Behavioral experiments on biased voting in networks. *Proc. Natl. Acad. Sci.* **106**, 1347–1352. https://doi.org/10.1073/pnas.0808147106 (2009).
7. Pak, J. & Zhou, L. A social network based analysis of deceptive communication in online chat. In *E-Life: Web-Enabled Convergence of Commerce, Work, and Social Life: 10th Workshop on E-Business, WEB 2011, Shanghai, China, December 4, 2011, Revised Selected Papers 10* 55–65. https://doi.org/10.1007/978-3-642-29873-8_6 (Springer, 2012).
8. Serrino, J., Kleiman-Weiner, M., Parkes, D. C. & Tenenbaum, J. Finding friend and foe in multi-agent games. *Adv. Neural Inf. Process. Syst.* **32**, 1 (2019).
9. Ibraheem, S., Zhou, G. & DeNero, J. Putting the con in context: Identifying deceptive actors in the game of mafia. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (eds. Carpuat, M. et al.) 158–168. https://doi.org/10.18653/v1/2022.naacl-main.11 (Association for Computational Linguistics, 2022).
10. Yu, F., Quartey, L. & Schilder, F. Exploring the effectiveness of prompt engineering for legal reasoning tasks. In *Findings of the Association for Computational Linguistics: ACL 2023* (eds. Rogers, A. et al.) 13582–13596. https://doi.org/10.18653/v1/2023.findings-acl.858 (Association for Computational Linguistics, 2023).
11. Wang, G. et al. Voyager: An open-ended embodied agent with large language models. Preprint at http://arxiv.org/abs/2305.16291 (2023).

12. Tan, W. et al. Cradle: Empowering foundation agents towards general computer control. Preprint at http://arxiv.org/abs/2403.03186 (2024).
13. Zhou, L. & Sung, Y.-W. Cues to deception in online Chinese groups. In *Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008)* 146–146. https://doi.org/10.1109/HICSS.2008.109 (IEEE, 2008).
14. Braverman, M., Etesami, O. & Mossel, E. Mafia: A theoretical study of players and coalitions in a partial information environment. *Ann. Appl. Probab.* **18**, 825–846 (2008).
15. Migdał, P. A mathematical model of the mafia game. Preprint at http://arxiv.org/abs/1009.1031 (2010).
16. Nakamura, N. et al. Constructing a human-like agent for the werewolf game using a psychological model based multiple perspectives. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)* 1–8. https://doi.org/10.1109/SSCI.2016.7850031 (IEEE, 2016).
17. Chuchro, R. Training an assassin AI for the resistance: Avalon. Preprint at http://arxiv.org/abs/2209.09331 (2022).
18. de Ruiter, B. & Kachergis, G. The mafiascum dataset: A large text corpus for deception detection. Preprint at http://arxiv.org/abs/1811.07851 (2018).
19. Lai, B. et al. Werewolf among us: Multimodal resources for modeling persuasion behaviors in social deduction games. In *Findings of the Association for Computational Linguistics: ACL 2023* (eds. Rogers, A. et al.) 6570–6588. https://doi.org/10.18653/v1/2023.findings-acl.411 (Association for Computational Linguistics, 2023).
20. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. Preprint at http://arxiv.org/abs/1810.04805 (2018).
21. Liu, Y. et al. Roberta: A robustly optimized bert pretraining approach. Preprint at http://arxiv.org/abs/1907.11692 (2019).
22. Radford, A. et al. Language models are unsupervised multitask learners. *OpenAI Blog* **1**, 9 (2019).
23. Vaswani, A. et al. Attention is all you need. In *Advances in Neural Information Processing Systems* (eds. Guyon, I. et al.), vol. 30 (Curran Associates, Inc., 2017).
24. Brown, T. et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems* (eds. Larochelle, H. et al.), vol. 33, 1877–1901 (Curran Associates, Inc., 2020).
25. Achiam, J. et al. Gpt-4 technical report. Preprint at http://arxiv.org/abs/2303.08774 (2023).
26. Touvron, H. et al. Llama: Open and efficient foundation language models. Preprint at http://arxiv.org/abs/2302.13971 (2023).
27. Wei, J. et al. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems* (eds. Koyejo, S. et al.), vol. 35, 24824–24837 (Curran Associates, Inc., 2022).
28. Yao, S. et al. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems* (eds. Oh, A. et al.), vol. 36, 11809–11822 (Curran Associates, Inc., 2023).
29. Kojima, T., Gu, S. S., Reid, M., Matsuo, Y. & Iwasawa, Y. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems* (eds. Koyejo, S. et al.), vol. 35, 22199–22213 (Curran Associates, Inc., 2022).
30. Loshchilov, I. & Hutter, F. Decoupled weight decay regularization. Preprint at http://arxiv.org/abs/1711.05101 (2017).
31. Kroeger, N., Ley, D., Krishna, S., Agarwal, C. & Lakkaraju, H. In-context explainers: Harnessing llms for explaining black box models. Preprint at http://arxiv.org/abs/2310.05797 (2024).
32. Renze, M. & Guven, E. The effect of sampling temperature on problem solving in large language models. Preprint at http://arxiv.org/abs/2402.05201 (2024).

Acknowledgements

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2019-0-01842, Artificial Intelligence Graduate School Program (GIST)).

Author contributions

B.Y. wrote the main manuscript and prepared figures and tables. K.K. reviewed the manuscript.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to K.-J.K.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by-nc-nd/4.0/.

© The Author(s) 2024