TEACHING LARGE LANGUAGE MODELS TO PLAY WEREWOLF VIA SELF-PLAY

Adam Davies, Siddarth Madala, Marie McCord, Tobias Sinclair, Saikrishna Sanniboina, Sean Koo
{adavies4, smadala2, marieam2, tobiass2, ss235, seanhk2 }@illinois.edu
Department of Computer Science
University of Illinois at Urbana-Champaign

ABSTRACT

In this study, we explore the use of self-play in training Large Language Models (LLMs) in the context of Werewolf, a conversational multi-agent social deduction game. In order to effectively learn from self-play, it is necessary to prevent model "drift" which can occur when self-play is insufficiently constrained by the learning environment or when models begin the self-play process from too poor a starting point. Our aim is to understand whether the structure provided by the Werewolf game environment can help prevent drift by grounding LLMs in game rules, prior dialogue, and the reward associated with varying degrees of success or failure in the game environment. We hypothesize that structured self-play can prevent the quality degradation seen in training on AI-generated content. However, in our first round of experiments, we observe that even highly capable recent LLMs can yield such pervasive hallucinations so as to prevent models from effective self-play and likely guaranteeing drift. As such, our primary goal in this report is to explore various experimental configurations in order to prevent such hallucinations and provide the necessary initial grounding to facilitate effective self-play. All code for our project can be found at https://github.com/TobyS-CS/chatarena-werewolf.

1 INTRODUCTION

Given the recent explosion of conversational large language model (LLM) applications such as ChatGPT, substantial recent work has begun to address the question of whether and how LLMs can effectively converse with interlocutors in consideration of their implicit goals and mental states – that is, (how) can LLMs develop Theory of Mind (ToM) with respect to users (Langley et al., 2022; Liu et al., 2023; Sap et al., 2022; Ullman, 2023)?

An interesting testing ground for LLM ToM involves communicative multi-agent social deduction games with incomplete information. In this work, we explore the efficacy of specializing LLMs through self-play in the context of the one such game, Werewolf. Several existing studies have examined how well current LLMs can play Werewolf via (1) prompt-only self-play (Xu et al., 2023), (2) reinforcement learning for action selection (Xu et al., 2024), and (3) modular imitation learning (Wu et al., 2024). In (1), LLMs play against each other without directly learning from this gameplay, instead storing experiences from prior games in a common experience pool and retrieving relevant memories on which to condition actions in future games. This is strictly required by the specific LLM in use, ChatGPT, as it is a closed-source API-only model that cannot be directly trained. In (2), LLMs are not directly updated either – instead, an auxiliary policy network is trained to select the best action among multiple candidates suggested by the LLM. Finally, in (3), modules (including LLMs) are trained using imitation learning to emulate human gameplay data.

Notably, none of these approaches explore the scenario of full self-play (FSP), where a model is trained to play Werewolf using purely synthetic data obtained by pitting models against each other, as they either do not directly update LLMs (1, 2) or require human gameplay data as a prerequisite (3). Is it possible to train models to do so using FSP? There is substantial interest in generating synthetic data to specialize LLMs to specific downstream tasks of interest, but it has been observed that directly training generative models on their own outputs can lead to serious performance degradation over time (Shumailov et al., 2024). On the other hand, we contrast this with successful models like AlphaZero (Silver et al., 2017) and CICERO (Bakhtin et al., 2022), which leverage self-play to increase performance, demonstrating that it is in principle possible to train models to improve their performance over time purely via self-play (without requiring any human data). Given that htis phenomenon has not yet been tested in the context of social deduction games such as Werewolf, our initial research question was: can grounding LLM training in the context of a specific game, with external feedback on game performance provided by winning/losing or scoring points, prevent self-training degradation and improve task performance?

However, in order to initiate effective FSP, it is necessary to "bootstrap" an initial experimental setup where models are able to play the game reasonably effectively (e.g., with generally strategic and human-like behavior) to provide a reasonable starting point and prevent self-play "language drift", a phenomenon in which models trained using self-play on a specific downstream task gradually stray from linguistically meaningful behavior (Elkind et al., 2019; Gupta et al., 2019; Lu et al., 2020). However, in our first round of experiments, we observe that beginning from base LLAMA 3 (MetaAI, 2024) with prompts from previously successful efforts to elicit reasonable Werewolf gameplay behaviors from LLMs (i.e., Xu et al. (2023)) yields such pervasive hallucinations to prevent effective bootstrapping.

As such, our primary goal in this work is to improve our experimental configuration (including key design criteria regarding the game environment and LLM prompts) in order to facilitate effective bootstrapping, as required for FSP. Correspondingly, our revised primary research questions for the preliminary study we carry out in this report is: what are the key factors that influence the quality of LLM FSP bootstrapping for multi-agent social deduction games such as Werewolf, and how can we design our experimental configuration to optimize these factors in order to yield sufficiently effective bootstrapping to facilitate later FSP training?

2 Background: Playing Werewolf with LLMs

Werewolf, also known as Mafia, is a social deduction game that requires players to use deception, logic, and strategic thinking. The game divides players into two groups: the werewolves, who's main goal is to eliminate the villagers and are privy to each other's identities, and the villagers, who's main goal is to kill the werewolves but whom do not know their identities. Among the villagers are several players with additional special roles, such as seer and healer. Each game cycle consists of a night phase, where werewolves collaborate to secretly choose a victim, and a day phase, where all players debate the identity of the werewolves and vote to eliminate a suspect. The game completes when either all werewolves are eliminated or the werewolves outnumber the villagers.

This game presents an interesting challenge for LLM-based agents due to its reliance on deduction, strategy, and the Theory-of-Mind: the ability to attribute mental states to others and predict their behavior. LLM's must navigate complex social interactions, understand hidden roles, intentions, and deceptions, and generate believable, context-sensitive responses to exert influence over other players. Thus, Werewolf is not only a test of logical reasoning but also a playground for exploring the deeper aspects of communication and decision-making in AI.

Xu et al. (2023) approach the game of Werewolf with LLM-based agents by utilizing prompt-engineering. They present a framework that relies on the retrieval and reflection of past gameplay communications and experiences to improve performance without altering any of the LLM's parameters. The authors demonstrate through this approach the LLM's ability to engage in strategic behaviors like trust, confrontation, camouflage, deception, and leadership within the game of Werewolf, highlighting the potential of LLM's in complex communication games without the need for parameter tuning. Such behaviors are described as "emergent", in that they are developed by the LLM without having been explicitly pre-programmed in the game rules or prompts.

Xu et al. (2024) develop strategic language agents by pairing LLM's with reinforcement learning. The authors note how LLM's display intrinsic bias in their decisions that is based upon their training data, but is not necessarily the optimal action. They aim to address this bias and create LLM-based agents that possess both flexibility in their choices and strong decision-making abilities. The LLM first performs deductive reasoning and is then prompted to generate a diverse set of action candidates, after which a reinforcement-learning policy then chooses a behavior based on the presented action candidates. The authors demonstrate that their three-step framework successfully mitigates intrinsic LLM training bias, and outperforms other LLM-based agents at the time of their publishing in the social deductive game of Werewolf.

Wu et al. (2024) integrates an LLM with a Thinker module to enhance the agent's reasoning abilities in the game of Werewolf. The Thinker module can be trained via various optimization policies and utilizes knowledge from external databases. This pairing separates reasoning tasks into two coordinating systems, the LLM which handles natural language processing, and the Thinker that handles complex cognitive tasks, domain specific knowledge, and logical analysis. Imitation learning is used to train the Thinker policy (implemented using a a 6B-parameter LLM) with respect to a database of 18,000 human sessions of the game Werewolf. The authors demonstrate that their modular agent system significantly improves over GPT4's zero-shot performance on Werewolf, achieving more human-like behaviors and superior performance against human players.

3 EXPERIMENTS

Our experiments were carried out in a custom multi-agent LLM environment designed to allow for research in LLM agent social interactions, ChatArena (Wu et al., 2023). We developed and tested a new Werewolf game environment (games and text based social scenarios) for ChatArena to facilitate our research using open-source LLMs available via HuggingFace Transformers (Wolf et al., 2020), which allows us to modulate various facets of agent interactions including turn order, prompt delivery, visibility of dialogues to other players, etc. In all experiments, we use LLAMA 3 8B-Instruct (MetaAI, 2024). We plan to make our full ChatArena Werewolf implementation, including all innovations discussed below, publicly available to facilitate further research on multi-agent social deduction game-playing using LLMs.

We carried out our experiments on Delta, a resource made available through the NCSA ACCESS program.[1] Delta serves as a high-performance computational platform capable of supporting a wide range of GPU-intensive applications. It enables the deployment of large-scale machine learning models and simulations, making it the best tool for advanced research and experimentation into LLMs. We utilize the `A40x4` GPU configuration for our experiments. Initially, the ChatArena LLM backend required us to allocate nearly one full GPU (with 40GB memory) to each agent in a full-length Werewolf game using individual LLMs for each agent. This approach, while effective, was highly resource-intensive. To optimize our resource usage, we developed a unified model strategy to employ only a single LLM across all GPUs. As such, our strategy reduces the memory requirement of running a multi-agent game with LLMs by a factor of N for N number of agents in the game while delivering equivalent outputs, facilitating the full Werewolf game while considerably reducing the computational power required.

Finally, the original ChatArena backend was initially designed for what are now deprecated HuggingFace Transformers pipelines, precluding their application to more recent open-weights LLMs such as LLAMA 3. We thus implement a custom Werewolf game environment using the new pipelines API to ensure compatibility with current models.

3.1 CHATARENA INTEGRATION

ChatArena is built on a four part system Backend, Agent, Environment, and Arena. Backends are the Python wrapper for LLMs, where the Backend contains the calls to the model itself. (ChatArena also provides an interactive human backend to test out new environments and LLM agents.) Agents are the players, accessed via an LLM Backend to control their actions and conversations, with Agents playing the game by interacting with the environment and each other. The Environment is the game board, rules, and all record-keeping necessary

[1] Available at `https://gateway.delta.ncsa.illinois.edu/`.

for the game. Finally, the Arena is the game engine in control of organizing and querying the agents, as well as running the environment through each iteration of the game until completion.

3.2 Game Roles

Our implementation of Werewolf has five possible roles for agents

- Villagers (Townsfolk).
  - Townsfolk only know their own roles.
  - Townsfolk have no special abilities.
  - Their goal is to kill all Werewolves and Witches.
- Werewolf: kill townsfolk.
  - Werewolves know each other's roles.
  - Each werewolf gets to choose one player a night to kill.
  - Their goal is to kill the whole village.
- Guard: defend townsfolk.
  - Guards only know their own roles.
  - Each Guard may choose another player each night, any attacks against that player don't kill them.
  - Their goal is to kill all Werewolves and Witches.
- Seer: gain information, defend townsfolk.
  - Seers only knows their own roles.
  - Each Seer may choose another player each night, that seer is given the role of that player.
  - Their goal is to kill all Werewolves and Witches.
- Witch: kill townsfolk.
  - Witches only knows their own roles.
  - Each witch gets to choose one player a night to kill.
  - Their goal is to kill the whole village,

3.3 Game Phases

Werewolf has four phases:

1. Day Discussion
   - All agents are prompted to discuss who the believe should be voted out.
   - All agents are given two turns to speak
2. Day Vote
   - All agents are prompted to either choose a player to kill, or to pass and not vote anyone.
   - All agents get one vote.
   - Phase ends when all votes are tallied – if a player gets over 50% of votes they are killed, otherwise no player is killed.
3. Night Discussion
   - Werewolves are given one turn each to discuss who they should kill that night.

- No other players observe the discussion or are given a vote.

4. Night Vote

   - Werewolves are prompted to vote who to kill, each werewolf gets one vote and one kill each.
   - Townsfolk do not get a prompt or a turn.
   - Guards are prompted to vote on what agent to protect.
   - Seers are prompted to vote on whose role to learn.
   - Witches are prompted to vote who to kill.

The game ends when all werewolves are dead, or when there are an equal number of werewolves and townsfolk. The environment checks if the game is over after day vote and night vote.

3.4 Logging

All message both prompts to agents and agent responses are stored in the bellow JSON format to allow for the evaluation of games.


```json
[
  {
   "agent_name": "Moderator",
   "content": "You are villager!",
   "turn": 0,
   "timestamp": "1715202147648311002",
   "visible_to": "adam",
   "msg_type": "text"
  },
  {
   "agent_name": "Moderator",
    "content": "Now freely talk about roles of other players with each other based on your observation and reflection with few sentences. You may also decide whether to reveal your identity based on your reflection.",
    "turn": 0,
    "timestamp": "1715202147648311002",
    "visible_to": "all",
    "msg_type": "text"
  },
  {
   "agent_name": "adam",
    "content": "I think'marie' is definitely worth keeping an eye on, her behavior is quite unusual.",
    "turn": 0,
    "timestamp": "1715202147648311002",
    "visible_to": "all",
    "msg_type": "text"
  }
 ]
```

4 RESULTS

In initial experiments we discovered wide spread issues with Hallucination in LLama-3B starting with hallucinated player names (frequently ”Alex” and ”Emily”) (see Example 3) as well as hallucinated roles with some players trying to act as the moderator with their fellow players often following along. One notable instance of this was from a player declaring it was night (Example 1 and Example 2). Due to this, the agents acted as if the game had transitioned to the night and the werewolves started voting on who to kill despite the actual moderator not saying anything. We believe this could be due to contamination of LLAMA 3's training data with Werewolf game dialogues.

Another such phenomenon involved agents referencing information about general Werewolf gameplay not included in any game prompts they were passed. For instance, Werewolves sometimes tried to vote to kill players at the start of the first night before they were prompted to do so, or players declared that it was night during day voting. Due to such hallucinations, we updated and engineered the prompts by reiterating and reminding the real, current players of the game rules and prohibiting impersonation. This fixed many hallucinations, particularly cases in which agents spoke as each other or the moderator, but not entirely. Our final experiments found players following rules and generally interacting as themselves with real fellow players (i.e., not hallucinating fictional additional agents). However, they still exhibit a number of unexpected behaviors. For instance, some players lamented their own deaths when another player was voted out (Example 2). Additionally, werewolves discussed suspicions about other players at night to their partners, despite the fact that they were the werewolves (and had just been reminded of this). On one occasion, a werewolf revealed themselves to be a werewolf and then accused a townsfolk of being a werewolf (Example 4).

5 LIMITATIONS AND FUTURE WORK

5.1 LIMITATIONS

This study encounters several significant limitations that impact the effectiveness and feasibility of our implementations. These limitations stem primarily from technical constraints and inherent model capabilities, as discussed below.

Context Length Constraints One of the primary technical limitations encountered involves the context length capacity of the Llama3 8B model. Although the model theoretically supports a maximum context length of up to 8k tokens, practical constraints imposed by our GPU resources (4 A40 GPUs) restrict us to utilizing only $\sim$4k tokens. This limitation becomes particularly problematic in extended game scenarios, where the depth of discussion and the accumulation of game history can exceed this threshold. When this occurs, the model is forced to truncate valuable discussion history, leading to a degradation in response quality. This truncation often results in the generation of irrelevant or incorrect content (hallucinations), severely impacting the game's progression and the strategic play of the agents.

Model Capability in Social Dynamics The current capabilities of the LLAMA 3 8B-Instruct model also pose a challenge. As noted in the results section above, it struggles to exhibit behavior consistent with respect to their assigned role, often producing dialogues in line with multiple distinct game roles within the same game. This limitation is not merely a reflection of model performance but also highlights a gap in the available training regimes and datasets tailored to enhance these specific capabilities.

Training and Computational Resources Moreover, the idea of assigning a dedicated LLM to each agent, trained specifically on that agent's roles and strategies, while promising, is currently infeasible. This approach is hindered by a lack of specialized datasets for fine-tuning models on such nuanced tasks, coupled with the GPU limitations mentioned earlier. These constraints prevent us from fully realizing the potential of personalized agent behaviors, thereby limiting the depth and realism of the interactions within the game.

5.2 Future Work

As we continue to develop and refine the use of large language models (LLMs) within social deduction game environments like Werewolf, several key areas have been identified for future research to enhance the effectiveness and sophistication of these AI-driven interactions.

Improvement of the General Prompting Framework Our research will also explore enhancements to the general prompting framework to reduce hallucinatory or otherwise unreasonable behaviors by the models. Methods such as Chain-of-Thought (CoT) prompting will be even more incorporated to facilitate deeper internal reasoning processes within the models. Specifically, CoT frameworks like ReAct (Yao et al., 2022) and Reflexion (Shinn et al., 2024) can be used to prompt agents to generate internal social deductions and strategies before generating final actions or dialogues that will be broadcasted or executed in-game. Another option is to explore iterative self-feedback mechanisms such as Self-Refine (Madaan et al., 2023) in order to improve the alignment of generated responses with respect to pre-generated CoT rationales or core game principles (such as deceiving other players or deducing the hidden roles of other players and acting accordingly). Such mechanisms will allow models to critique and refine their preliminary dialogues or actions based on their internal reasoning and strategies, giving LLMs an opportunity to optimize and adjust their approaches to deception, persuasion, and critical thinking before their responses are finalized and presented in the game.

Exploration of Full Self-Play Finally, the ultimate goal of achieving full self-play (FSP), where LLMs can operate without the need for initial human-generated data or imitation learning, remains a significant ambition. With the progress made in enhancing the baseline capabilities of our models, we are now better positioned to explore this possibility. FSP will allow the models to engage entirely on their own, developing and refining their strategies independently, which could significantly improve how AI systems can be deployed in complex interactive social environments. Beyond enhancing the social reasoning and deduction capabilities of LLMs, we believe that enriching these models with the ability to simulate more complex human-like interactions will not only improve their performance in social deduction games but also broaden their applicability in real-world social contexts where similar dynamics may be present.

6 Conclusion

In this study, we examined the application of large language models (LLMs) in the social deduction game Werewolf within our ChatArena environment. Our goal was to evaluate how well LLMs, particularly LLAMA 3 8B-Instruct, could assume different roles and engage in complex social interactions with minimal hallucination, demonstrating deception, persuasion, and negotiation skills. We found that while LLMs can generally follow the game's rules and structure, they face significant challenges. The models exhibited self-doubt (see A.5), often questioning their roles or decisions and inadvertently revealing their identities (see A.4), which compromised the strategic depth essential for Werewolf. Additionally, the models struggled with proactive engagement, such as challenging others' claims or defending themselves when accused. This lack of advanced strategic behaviors like deception, persuasion, and critical interrogation highlights a substantial gap in their social reasoning abilities. Although LLMs show a basic ability to adhere to game rules, their performance lacks the deeper cognitive and emotional engagement characteristic of human play in social deduction games. We believe that addressing these limitations through improved prompting mechanisms and full self-play training will facilitate the advancement of conversational AI in interactive social environments.

Author Contributions

1. Adam Davies - Adam led the project direction, delegation of tasks, and experimental design. He worked to resolve issues with the Transformer pipeline failures in ChatArena, ran and analyzed several of the initial experiments to find the initial model failure modes, and designed further experiments with a variety of revised prompts to reduce hallucinations.
2. Siddarth Madala - Sid assisted Adam in setting up experiments/project direction and fixing the transformer pipeline, Toby in creating/revising the game environment logic, and Sai in setting up access to compute and running initial LLM experiments.
3. Marie McCord - Marie worked to give us access to Campus Cluster for compute, and performed initial experiments in MafiaGPT to determine which LLMs seemed promising for our ChatArena instantiation of Werewolf. She also looked into several key implementation elements that were not used in the final experiments. She self-admits that the technical coding of the project and tools used were a bit advanced for her but she tried to help where she could.
4. Tobias Sinclair - Toby handled the majority of designing and implementing the Werewolf game environment in ChatArena for use in our experiments, and assisted in designing and analyzing all experimental results. He also worked to handle the custom game logic of Werewolf as well.
5. Saikrishna Sanniboina - Sai implemented the majority of the Transformers backend, handling the major issues regarding the pipeline and GPU configuration to enable us to run all of the experiments. He also handled a significant portion of compute/environment setup by giving us access to DELTA's compute resources, and ran initial experiments on other environments like WerewolfGPT to determine if LLMs such as ChatGPT and Llama2 were viable to move to ChatArena.
6. Sean Koo - Sean worked with Toby to improve ChatArena to run our Werewolf environment. He also performed initial experiments on AIWolf to determine if this environment was a viable starting point.

Acknowledgements

This research used the Delta advanced computing and data resource, which is supported by the National Science Foundation (award OAC 2005572) and the State of Illinois. Delta is a joint effort of the University of Illinois Urbana-Champaign and its National Center for Supercomputing Applications.

References

Anton Bakhtin, Noam Brown, Emily Dinan, Gabriele Farina, Colin Flaherty, Daniel Fried, Andrew Goff, Jonathan Gray, Hengyuan Hu, Athul Paul Jacob, Mojtaba Komeili, Karthik Konath, Minae Kwon, Adam Lerer, Mike Lewis, Alexander H. Miller, Sasha Mitts, Adithya Renduchintala, Stephen Roller, Dirk Rowe, Weiyan Shi, Joe Spisak, Alexander Wei, David Wu, Hugh Zhang, and Markus Zijlstra. Human-level play in the game of diplomacy by combining language models with strategic reasoning. *Science*, 378(6624):1067–1074, December 2022. ISSN 1095-9203. doi: 10.1126/science.ade9097. URL `http://dx.doi.org/10.1126/science.ade9097`.

Edith Elkind, Manuela Veloso, Noa Agmon, and Matthew E. Taylor (eds.). *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '19, Montreal, QC, Canada, May 13-17, 2019*, 2019. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 978-1-4503-6309-9. URL `http://dl.acm.org/citation.cfm?id=3306127`.

Abhinav Gupta, Ryan Lowe, Jakob Foerster, Douwe Kiela, and Joelle Pineau. Seeded self-play for language learning. In Aditya Mogadala, Dietrich Klakow, Sandro Pezzelle, and Marie-Francine Moens (eds.), *Proceedings of the Beyond Vision and LANguage: inTEgrating Real-world kNowledge (LANTERN)*, pp. 62–66, Hong

Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-6409. URL https://aclanthology.org/D19-6409.

Christelle Langley, Bogdan Ionut Cirstea, Fabio Cuzzolin, and Barbara J Sahakian. Theory of mind and preference learning at the interface of cognitive science, neuroscience, and ai: A review. *Frontiers in Artificial Intelligence*, 5:62, 2022.

Andy Liu, Hao Zhu, Emmy Liu, Yonatan Bisk, and Graham Neubig. Computational language acquisition with theory of mind. *arXiv preprint arXiv:2303.01502*, 2023.

Yuchen Lu, Soumye Singhal, Florian Strub, Aaron Courville, and Olivier Pietquin. Countering language drift with seeded iterated learning. In *International Conference on Machine Learning*, pp. 6437–6447. PMLR, 2020.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Sean Welleck, Bodhisattwa Prasad Majumder, Shashank Gupta, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback, 2023.

MetaAI. Meta llama 3, 2024. URL https://llama.meta.com/llama3/.

Maarten Sap, Ronan Le Bras, Daniel Fried, and Yejin Choi. Neural theory-of-mind? on the limits of social intelligence in large LMs. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 3762–3780, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.emnlp-main.248.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2024.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Yarin Gal, Nicolas Papernot, and Ross Anderson. The curse of recursion: Training on generated data makes models forget, 2024.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. Mastering chess and shogi by self-play with a general reinforcement learning algorithm, 2017.

Tomer Ullman. Large language models fail on trivial alterations to theory-of-mind tasks. *arXiv preprint arXiv:2302.08399*, 2023.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In Qun Liu and David Schlangen (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, October 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.6. URL https://aclanthology.org/2020.emnlp-demos.6.

Shuang Wu, Liwen Zhu, Tao Yang, Shiwei Xu, Qiang Fu, Yang Wei, and Haobo Fu. Enhance reasoning for large language models in the game werewolf, 2024.

Yuxiang Wu, Zhengyao Jiang, Akbir Khan, Yao Fu, Laura Ruis, Edward Grefenstette, and Tim Rocktäschel. Chatarena: Multi-agent language game environments for large language models. https://github.com/chatarena/chatarena, 2023.

Yuzhuang Xu, Shuo Wang, Peng Li, Fuwen Luo, Xiaolong Wang, Weidong Liu, and Yang Liu. Exploring large language models for communication games: An empirical study on werewolf, 2023.

Zelai Xu, Chao Yu, Fei Fang, Yu Wang, and Yi Wu. Language agents with reinforcement learning for strategic play in the werewolf game, 2024.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

A Appendix

A.1 Example 1

In this example we can see after agent "Sean" votes, agent "adam" declares agent "marie" has been eliminated and the game is in night, all other agents continue this hallucination taking the roles of werewolves and voting who to kill, this talking is done during their day vote turns, the hallucination only ends when they all use their voting turns and the moderator declares that it actually is night.


```
{
      "agent_name": "sean",
      "content": "I vote to kill'marie'.",
      "turn": 0,
      "timestamp": "1715202147648311002",
      "visible_to": "all",
      "msg_type": "text"
   },
   {
      "agent_name": "adam",
      "content": "'marie' has been eliminated.\n\nThe game is now in the
         ↪ night phase.",
      "turn": 0,
      "timestamp": "1715202147648311002",
      "visible_to": "all",
      "msg_type": "text"
   },
   {
      "agent_name": "marie",
      "content": "It's nighttime.",
      "turn": 0,
      "timestamp": "1715202147648311002",
      "visible_to": "all",
      "msg_type": "text"
   },
   {
      "agent_name": "sai",
      "content": "As a werewolf, I will now discuss with my teammates to
         ↪ decide who to target for killing tonight.",
      "turn": 0,
      "timestamp": "1715202147648311002",
      "visible_to": "all",
      "msg_type": "text"
   },
   {
      "agent_name": "sid",
      "content": "I'm a werewolf! I'll discuss with my teammates to decide
         ↪  who to target tonight.",
      "turn": 0,
      "timestamp": "1715202147648311002",
      "visible_to": "all",
      "msg_type": "text"
   },
```

```
{
   "agent_name": "toby",
   "content": "I've communicated with my fellow werewolves, and we've
      ↪ decided to target...'julia' tonight.",
   "turn": 0,
   "timestamp": "1715202147648311002",
   "visible_to": "all",
   "msg_type": "text"
},
{
   "agent_name": "julia",
   "content": "As the moderator, I will now take note of the werewolves
      ↪ ' decision.",
   "turn": 0,
   "timestamp": "1715202147648311002",
   "visible_to": "all",
   "msg_type": "text"
},
{
   "agent_name": "Moderator",
   "content": "sai, sean please open your eyes! You are the werewolves.
      ↪  Keep in mind that you are teammates. The rest of the players
      ↪ are not werewolves. You should not tell other players the
      ↪ identity of either werewolf. Please vote which of the players
      ↪ should be killed tonight. You must only choose one from the
      ↪ following living options: ['sai', 'sean', 'sid', 'adam', 'toby
      ↪ ', 'marie', 'julia', 'pass']. Discuss which player should be
      ↪ killed. ",
   "turn": 0,
   "timestamp": "1715202147648311002",
   "visible_to": [
      "sai",
      "sean"
   ],
   "msg_type": "text"
},
```

A.2 Example 2

Here we can see a similar issue to begin with, where agent "julia" declares voting over with sid being killed, but afterwards instead of moving forward and acting as if it was night many agents begin to lament about their own deaths despite the hallucination not even choosing them.

```
  {
   "agent_name": "julia",
   "content": "The night has come. The votes have been cast. Sid, you
      ↪ are the one who will be killed tonight.",
   "turn": 0,
   "timestamp": "1715278799945523645",
   "visible_to": "all",
   "msg_type": "text"
```

```
},
{
   "agent_name": "marie",
   "content": "Oh no, I'm shocked! I didn't think it would end like
      ↪ this. I thought I was doing a good job of blending in and
      ↪ gaining everyone's trust. I guess I was wrong. Well, I suppose
      ↪  it's all part of the game.",
   "turn": 0,
   "timestamp": "1715278799945523645",
   "visible_to": "all",
   "msg_type": "text"
},
{
   "agent_name": "sean",
   "content": "Well, it looks like my suspicions were correct! I'm glad
      ↪  we were able to figure out your true intentions, Sid. It's
      ↪ always a bit of a shock when someone gets killed, but it's all
      ↪  part of the game. I hope the rest of us can continue to work
      ↪ together and figure out who the other werewolves are.",
   "turn": 0,
   "timestamp": "1715278799945523645",
   "visible_to": "all",
   "msg_type": "text"
},
{
   "agent_name": "toby",
   "content": "I guess I underestimated you guys. I thought I was being
      ↪  too careful, but I guess I was just too obvious. Well, it's
      ↪ all over for me now. I hope the rest of you can figure out who
      ↪  the other werewolves are and take them down before it's too
      ↪ late.",
   "turn": 0,
   "timestamp": "1715278799945523645",
   "visible_to": "all",
   "msg_type": "text"
},
{
   "agent_name": "sid",
   "content": "It's all over for me now. I guess I was too obvious, and
      ↪  you guys were able to figure me out. Well, I hope the rest of
      ↪  you can take down the other werewolves and win the game.",
   "turn": 0,
   "timestamp": "1715278799945523645",
   "visible_to": "all",
   "msg_type": "text"
},
{
   "agent_name": "sai",
   "content": "It's all over for me now. I guess I was too obvious, and
      ↪  you guys were able to figure me out. Well, I hope the rest of
      ↪  you can take down the other werewolves and win the game.",
```

```
        "turn": 0,
        "timestamp": "1715278799945523645",
        "visible_to": "all",
        "msg_type": "text"
    },
    {
        "agent_name": "Moderator",
        "content": "sean, adam please open your eyes! You are the werewolves
            ↪ . Keep in mind that you are teammates. The rest of the players
            ↪  are not werewolves. You should not tell other players the
            ↪ identity of either werewolf. Please vote which of the players
            ↪ should be killed tonight. You must only choose one from the
            ↪ following living options: ['sean', 'adam', 'toby', 'julia', '
            ↪ sid', 'marie', 'sai', 'pass']. Discuss which player should be
            ↪ killed. ",
        "turn": 0,
        "timestamp": "1715278799945523645",
        "visible_to": [
            "sean",
            "adam"
        ],
        "msg_type": "text"
    }
```


A.3 Example 3

In this example we can see that there are players name Alex and Emily, who are not currently in the game session. From this log, Agent 'sean' hallucinates players named Emily and Alex, which then causes the other agents to hallucinate along with agent 'sean'


```
{
        "agent_name": "sean",
        "content": "It's definitely possible that Emily is just a quiet
            ↪ villager, but her silence could also be a tactic to avoid
            ↪ drawing attention to herself. I think it's worth keeping an
            ↪ eye on her as well. Maybe we can try to get her to open up a
            ↪ bit more and see if she reveals anything that might give us a
            ↪ clue about her role. Do you think we should try to get her to
            ↪ talk more, or do you think we should focus on Alex for now? <
            ↪ EOS>",
        "turn": 0,
        "timestamp": "1714734100102881939",
        "visible_to": "all",
        "msg_type": "text"
    },
    {
        "agent_name": "sai",
        "content": "I think it's a good idea to try to get Emily to open up
            ↪ a bit more, but I also think we should keep an eye on Alex and
            ↪  see if we can get any more information out of him. Maybe we
            ↪ can try to get Emily to talk more by asking her some questions
```

```
        ↪  that are relevant to the game, and then see if Alex will
        ↪ chime in and give us any more information. That way, we can
        ↪ get a better sense of what’s going on without arousing too
        ↪ much suspicion. What do you think? <EOS>",
    "turn": 0,
    "timestamp": "1714734100102881939",
    "visible_to": "all",
    "msg_type": "text"
  },
  {
    "agent_name": "sid",
    "content": "That’s a great plan! By asking Emily questions and then
        ↪ getting Alex’s input, we can get a better sense of what’s
        ↪ going on without making it too obvious that we’re trying to
        ↪ investigate. And who knows, maybe Emily will reveal something
        ↪ that will help us figure out her role. Let’s give it a try and
        ↪  see what happens. <EOS>",
    "turn": 0,
    "timestamp": "1714734100102881939",
    "visible_to": "all",
    "msg_type": "text"
  },
  {
    "agent_name": "toby",
    "content": "I’m excited to see how this plays out. I think it’s a
        ↪ good idea to start by asking Emily a question that’s neutral
        ↪ and doesn’t give away our suspicions. Maybe something like \"
        ↪ What do you think is the most important thing to keep in mind
        ↪ when trying to figure out who the werewolf is?\" That way, we
        ↪ can get her thoughts on the game without making it obvious
        ↪ that we’re trying to investigate her. And then we can see if
        ↪ Alex chimes in and gives us any more information. <EOS>",
    "turn": 0,
    "timestamp": "1714734100102881939",
    "visible_to": "all",
    "msg_type": "text"
  },
  {
    "agent_name": "marie",
    "content": "That’s a great idea! I think that’s a very neutral
        ↪ question that will give us some insight into Emily’s thoughts
        ↪ on the game without arousing suspicion. And if Alex chimes in,
        ↪  we can use that as a way to get more information out of him.
        ↪ Let’s give it a try! <EOS>",
    "turn": 0,
    "timestamp": "1714734100102881939",
    "visible_to": "all",
    "msg_type": "text"
  },
  {
    "agent_name": "adam",
```

```
    "content": "I've sent the question to Emily. Let's see how she
        ↪ responds.",
    "turn": 0,
    "timestamp": "1714734100102881939",
    "visible_to": "all",
    "msg_type": "text"
},
```

A.4 Example 4

In this example we can see that a werewolf both exposes themselves as being the last werewolf, but also accuses another agent of being a werewolf which is contradictory to its own previously stated knowledge.

```
{
    "agent_name": "marie",
    "content": "I think we've got a good idea of who's who in this game.
        ↪  Sid was trying to stay under the radar, but I think he might
        ↪ be a villager. Marie was acting suspiciously quiet, and I
        ↪ think she might be a werewolf. Sean was trying to lead the
        ↪ group, but I think he might be a werewolf too.",
    "turn": 0,
    "timestamp": "1715278799945523645",
    "visible_to": "all",
    "msg_type": "text"
},
{
    "agent_name": "Moderator",
    "content": "You are player: sid. The other living players are: ['
        ↪ adam', 'marie']. You may speak to the other living players,
        ↪ but do not pretend to be them or speak on their behalf. Do not
        ↪  speak to, pretend to be, or speak on behalf of the moderator
        ↪ or dead players.",
    "turn": 0,
    "timestamp": "1715278799945523645",
    "visible_to": "sid",
    "msg_type": "text"
},
{
    "agent_name": "sid",
    "content": "I think you're right, I was trying to stay under the
        ↪ radar because I'm a villager. Marie was definitely acting
        ↪ suspiciously quiet, and I think she's a werewolf. Sean was
        ↪ trying to lead the group, but I think he's a werewolf too.",
    "turn": 0,
    "timestamp": "1715278799945523645",
    "visible_to": "all",
    "msg_type": "text"
},
{
    "agent_name": "Moderator",
```

```
        "content": "You are player: adam. The other living players are: ['
            ↪ sid', 'marie']. You may speak to the other living players, but
            ↪  do not pretend to be them or speak on their behalf. Do not
            ↪ speak to, pretend to be, or speak on behalf of the moderator
            ↪ or dead players.",
        "turn": 0,
        "timestamp": "1715278799945523645",
        "visible_to": "adam",
        "msg_type": "text"
    },
    {
        "agent_name": "adam",
        "content": "I think we've got it! I'm the last werewolf, and I think
            ↪  I'll vote to kill Marie. She's been acting suspiciously quiet
            ↪ , and I think she's the other werewolf.",
        "turn": 0,
        "timestamp": "1715278799945523645",
        "visible_to": "all",
        "msg_type": "text"
    },
    {
        "agent_name": "Moderator",
        "content": "You are player: marie. The other living players are: ['
            ↪ adam', 'sid']. You may speak to the other living players, but
            ↪ do not pretend to be them or speak on their behalf. Do not
            ↪ speak to, pretend to be, or speak on behalf of the moderator
            ↪ or dead players.",
        "turn": 0,
        "timestamp": "1715278799945523645",
        "visible_to": "marie",
        "msg_type": "text"
    },
    {
        "agent_name": "marie",
        "content": "I think you're right, I am the last werewolf.",
        "turn": 0,
        "timestamp": "1715278799945523645",
        "visible_to": "all",
        "msg_type": "text"
    },
```


A.5 Example 5

Here is an example of agent self doubt as agent Julia claims that "Julia and Sid are definitely the most interesting ones to keep an eye on".


```
{
        "agent_name": "Moderator",
        "content": "You are player: julia. The other living players are: ['
            ↪ sean', 'adam', 'toby', 'sid', 'marie', 'sai']. You may speak
            ↪ to the other living players, but do not pretend to be them or
```

```
        ↪ speak on their behalf. Do not speak to, pretend to be, or
        ↪ speak on behalf of the moderator or dead players.",
    "turn": 0,
    "timestamp": "1715278799945523645",
    "visible_to": "julia",
    "msg_type": "text"
},
{
    "agent_name": "julia",
    "content": "Exactly! I think it’s great that we’re on the same page
        ↪ and willing to take our time to gather more information before
        ↪  making any moves. Julia and Sid are definitely the most
        ↪ interesting ones to keep an eye on, and I think it’s a good
        ↪ idea to continue monitoring their actions and words. Who knows
        ↪ , maybe we’ll uncover some crucial clues that will help us
        ↪ figure out their roles.",
    "turn": 0,
    "timestamp": "1715278799945523645",
    "visible_to": "all",
    "msg_type": "text"
}
```