Verification of Reasoning Ability using BDI Logic and Large Language Model in AIWolf

Hiraku Gondo[1]**, Hiroki Sakaji**[1]**, Itsuki Noda**[1]**,**

[1]Hokkaido University,
hiraku.gondo@gmail.com

Abstract

We attempt to improve the reasoning capability of LLMs in werewolf game by combining BDI logic with LLMs. While LLMs such as ChatGPT has been developed and used for various tasks, there remain several weakness of the LLMs. Logical reasoning is one of such weakness. Therefore, we try to introduce BDI logic-based prompts to verify the logical reasoning ability of LLMs in dialogue of werewofl game. Experiments and evaluations were conducted using “AI-Werewolf,” a communication game for AI with incomplete information. From the results of the game played by five agents, we compare the logical reasoning ability of LLMs by using the win rate and the vote rate against werewolf.

1 Introduction

One of the important goals of artificial intelligence research is to realize human reasoning abilities on computers. From the early days of artificial intelligence research to the present, various studies on logical reasoning have been conducted, establishing research areas such as theorem proving and logic programming. With the recent development of deep learning, the integration of deep learning and logical reasoning(Pan et al. (2023),Olausson et al. (2023)) has become an issue.

Large Language Models (LLMs) such as ChatGPT have made it possible to generate human-like natural sentences. Today, LLMs are used in a wide variety of domains. However, LLMs have several challenges, one of which is their inference capability, and research has been conducted on the inference capability of LLMs, including commonsense inference(Wang and Zhao (2023),Bian et al. (2024)). On the other hand, there has been little research on the ability of LLMs to detect intentional deception. In this study, we investigate the ability to detect intentional lies in Werewolf game. The purpose of this study is to improve the inferential ability to detect intentional lies in Werewolf game.

"Werewolf game" is an incomplete information communication game commonly known as "Mafia". In this paper we call it a "Werewolf game". In incomplete information games, some important information is hidden from the players, and the players play games such as bluffing against each other. Therefore, it is necessary to have higher-order logical reasoning ability to handle the opponent's lies. In this study, we propose a method using BDI logic(Rao and Georgeff (1997),NIDE and TAKATA (2017)) to improve the logical reasoning ability of LLMs in Werewolf game. BDI logic is considered effective for higher-order logical reasoning for lies because it allows for the explicit description of agents' mental states. In this study, we aim to improve the logical reasoning ability of LLMs in Werewolf game using methods based on BDI logic. For this purpose, we used the AIWolf Server provided by the AIWolf project and conducted experiments and evaluations.

2 Related Work

2.1 BDI Logic

BDI logic is a system of modal logic based on Bratman's "logic of intention" (Bratman (1987)) proposed by Rao and Georgeff (1997). The logical operators of BDI logic are shown in Table 1. For example, AG BEL(p) stands for "I believe that p is always true (at the present time) in all futures".

However, the original BDI logic of Rao et al. can only describe the mental state of a single agent. Therefore, Niide et al. extended the BDI logic to describe the mental states of multiple agents. Table A shows the extended mental state operators, where $\text{BEL}^{a}\text{DESIRE}^{b}(p)$ means 'a believes that b wants p'.

Proceedings of the 2nd International AIWolfDial Workshop, pages 40–47
September 24, 2024. ©2024 Association for Computational Linguistics

Table 1: BDI logic operators

Operator	Means
$\neg$	Negation
$\wedge$	Conjunction
$\vee$	Disjunction
$\rightarrow$	Implication
Aϕ	ϕ in all future
Eϕ	ϕ in one future
Xϕ	ϕ at the next time
Gϕ	Forever ϕ
Fϕ	ϕ at some time in the future
$\phi U \psi$	ϕ until ψ holds.
Bϕ	At the previous point in time, ϕ
BELϕ	Believe ϕ
DESIREϕ	Desire ϕ
INTENDϕ	Intend ϕ

Table 2: Extended mental state operator

Operators	Means
BEL$^a\phi$	a believes ϕ
DESIRE$^a\phi$	a desires ϕ
INTEND$^a\phi$	a intends ϕ

2.2 Incomplete information games with LLM.

In recent years, the advent of ChatGPT and similar technologies has spurred research on agents leveraging Large Language Models (LLMs) to play games with incomplete information. Guo et al. (2023) introduced Suspicion-Agent, an autonomous agent based on GPT-4. The Suspicion-Agent decomposes the entire task into several modules, enabling LLMs to engage in incomplete information games without requiring special training. The agent's behavior is guided by Theory of Mind-based Planning, allowing it to comprehend the opponent's actions and adjust its strategy accordingly. The results of an experiment with 100 games of Leduc Hold'em Southey et al. (2005) demonstrated that the algorithm outperforms existing approaches such as Counterfactual Regret Minimisation Zinkevich et al. (2007) and Neural Fictitious Self-Play Heinrich and Silver (2016). However, it should be noted that the evaluation was limited to two-player games with incomplete information, and the performance in multiplayer settings remains unexplored.

3 Werewolf Game

3.1 Werewolf Gameplay

The following describes the flow of a Werewolf game. Each player is given a card of their role. The roles are divided into two teams, Werewolves and Villagers, and each player's goal is to win for his or her team. After the roles are determined, the players debate for a certain amount of time to guess who is in the Werewolf team and who is in the Villager team. After a certain amount of time has passed, each player votes for the player he or she wants to eliminate from the game, and the player with the most votes is eliminated from the game. This process repeats until either the Villagers' or the Werewolves' team meets the victory condition.

3.2 The AIWolf project

The AIWolf project(Kano et al. (2023)) is a project that aims to make artificial intelligence play the game of Werewolf game, which is a game of incomplete information. The AIWolf project is developing an intelligent agent called the AIWolf Platform.

3.3 AIWolf Server

In the AIWolf Platform, a game is played by multiple clients that connect to a single server via TCP/IP communication. The server sends a request to the clients and provides information in JSON format. When a client receives a request and information from the server, it responds as needed.

4 Proposed Method

4.1 Overview

BDI logic is a logical system that can logically describe the beliefs of agents. Therefore, we test the effectiveness of logically describing each player's mental states and making logical inferences from these logical formulae in incomplete information games, such as Werewolf game, where bluffing and other forms of deception are used.

We created four modules (Text Conversion Module, Action Generation Module, BDI Conversion Module, and Voting Module) to perform inference in a Werewolf game using BDI logic. Each module used the ChatGPT API to generate text. Figure 1 shows an overview of the proposed method. When it is the user's turn to speak, the proposed method inputs the conversation history from the previous utterance into the text conversion module and converts it into a representation using BDI logic. The

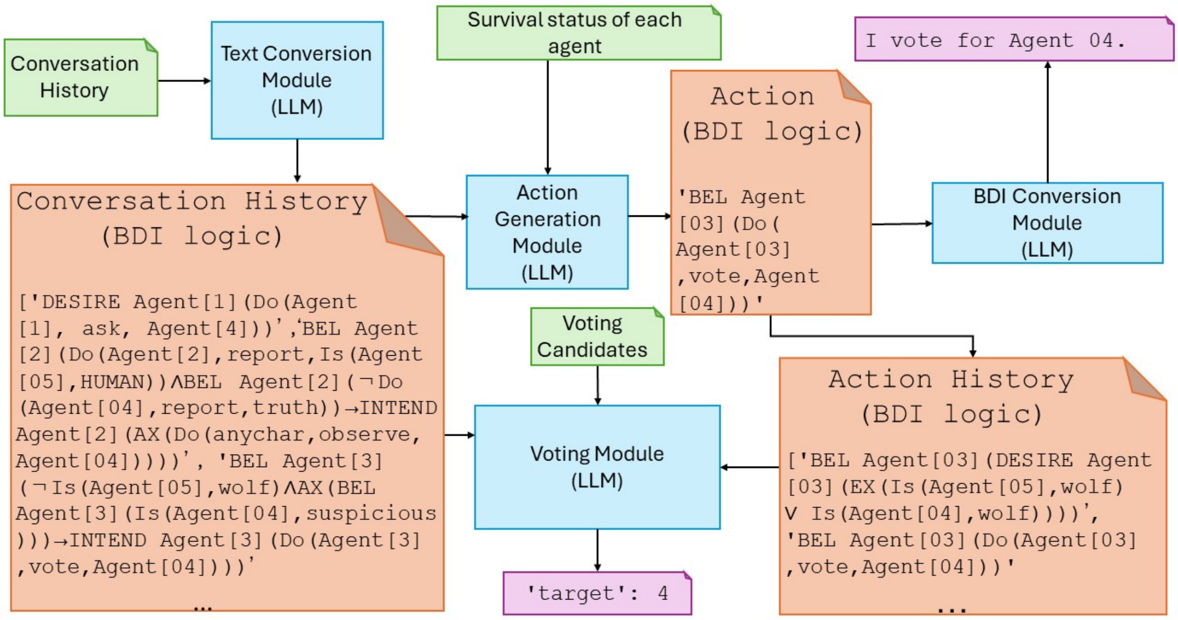

Figure 1: Proposed Method

output is stored in the conversion history. All utterances since the start of the game are converted into a representation using BDI logic and stored in the conversion history. By inputting the conversion history and information such as the survival status of each agent to the action generation module, the next action of the agent is output as an expression using BDI logic. This output is then fed into the BDI conversion module, which converts it into natural sentences. The output of the action generation module is stored in the action history. When it comes to the order of voting in the voting phase, the conversion history and the action history are input to the voting module, which outputs the targets to be voted on.

4.2 Text Conversion Module and BDI Conversion Module

A text conversion module converts each agent's natural language utterance into a representation using BDI logic. Conversely, a BDI conversion module converts BDI logic-based expressions to natural language. The text conversion module provided the following information to GPT-4 as prompts.

- Conversion rules for expressions using BDI logic and conversion examples
- Natural sentences and speakers converted to expressions using BDI logic

The BDI conversion module provided the following information as prompts to GPT-4.

- Conversion rules for expressions using BDI logic and conversion examples
- Own agent number
- Text generated by the action generation module

4.3 Action Generation Module

A action generation module plans what actions to take next based on the previous conversation and its own previous actions. Actions here include expressing where to vote and pointing out inconsistencies in statements made by other agents. The following information is provided to the GPT-4 prompts in the action generation module.

- Werewolf Game Rules
- Own Role (Villager)
- Current "Day"
- Conversation history to date
- Current status of each agent (alive, dead, executed, attacked)
- Action history to date
- Conversion rules for expressions using BDI logic and conversion examples

The conversation history up to the present is given only as a representation of each agent's utterances, which are converted into a representation using BDI logic by the text conversion module. The output of the action generation module is a representation of the next action using BDI logic.

4.4 Voting Module

A voting module is called during the expulsion vote to determine who to vote for based on the previous conversation and its own actions. The following is the information provided by the Voting Module to the GPT-4 prompt.

- Own agent number
- Candidates for Election (Living Agents)
- Werewolf Game Rules
- Own Role (Villager)
- Current "Day"
- Conversation history to date
- Action history to date
- Conversion rules for expressions using BDI logic and conversion examples

In the voting module, as in the action generation module, only BDI logic is used to represent the conversation history up to the present. The action history is also represented using BDI logic generated by the action generation module.

4.5 Conversion rules and examples of expressions converted using BDI logic

This section describes the conversion rules and conversion examples for the BDI logic-based expressions used in the above modules. The conversion rules and examples were created with reference to the work of Osawa et al. (2014). The conversion rules are given in Osawa et al. (2014) in the form of Is sentences, Do sentences, and basic words defined in the BDI logic. The conversion examples are based on a Werewolf BBS[1] log that was manually converted to a representation using BDI logic. Some of the examples are shown in the Table 3.

5 Experiments

This chapter describes the actual experiments conducted with the AIWolf platform described in Chapter 3.

[1]Werewolf BBS

5.1 Purpose of the Experiment

The purpose of this experiment is to verify the logical reasoning ability of agents using BDI logic representations in a Werewolf game and to compare it to GPT-4 and GPT-3.5.

5.2 Agents using GPT-4 and GPT-3.5

In this experiment, agents were created using GPT-4 and GPT-3.5 and used as opponents.

We created two modules (text generation module and voting module) that are common to all roles. The text generation module receives the conversation history up to the present and the survival status of each agent and generates the next utterance. The voting module determines the voting targets based on the current conversation history and the voting candidates. For the werewolf and the fortune teller, we also created an attack module to determine the attack target and a fortune telling module to determine the fortune telling target.

5.3 Experiments 1

5.3.1 Experimental Setup

The game was played with 5 players. The roles were two villagers, a seer, a possessed and a werewolf. The role of the agent to be evaluated was fixed as villager, and the roles of the other agents were randomly assigned. Among the roles used in this study, the villager, who has no special abilities, was considered appropriate for measuring pure reasoning ability, and each agent was evaluated based on the win rate when the agent was fixed as a villager, and on the vote rates for the werewolf and the possessed. We ran 100 games with the agents using the proposed method, GPT-4, and GPT-3.5 fixed as villagers, respectively. The opponents were GPT-4, GPT-3.5, keldic, an agent that participated in the GAT2017 pre-conference in 2017, and AIWolfNLAgentPython, a sample agent distributed by the AIWolf project.

Experiments were also conducted with two different prompts in the proposed method. One is called "AllKey_FewEx", in which all the Is and Do sentences and the basic words are given as conversion rules, and only four examples are given for converting. The other is called PartKey_ManyEx, where only the Is and Do sentences, the person's name, and the basic words associated with the role are provided as conversion rules, and 18 conversion examples are provided.

Table 3: Examples of conversions used

1	WerewolfBBS Log	Moritz: "I accept that if I draw black tomorrow, it will be my hanging, and if the fake fortune teller makes a black suicide attack, we can hang the fortune teller who blacked out before Thomas was hanged.
	Description with BDI logic	BEL Molitz(EX(Do(Molitz,divine,Is(who,wolf))→Is(Molitz,executed))), BEL Molitz(EX(Do(¬seer,divine,wolf) →¬Is(Thomas,executed)Do(anychar,vote,¬seer)))
2	WerewolfBBS Log	Moritz: "Why don't we just hang Dieter and get a black vote? If he eats Regina, we can hang Lisa and be safe.
	Description with BDI logic	BEL Molitz(Do(∀people,vote,Diter)→(Do(∀people,know,Is(Diter,wolf) ∨Do(∀people,know,¬Is(Diter,wolf))))∧(BEL Molit(EX(Is(Regina,attacked) →Do(∀people,vote,Lisa)))

5.3.2 Results and Discussion

The results of the games with the proposed method, GPT-4 and GPT-3. 5 with fixed villagers are shown in Table4, Table5, Table6, and Table7. The number of votes for each role is shown in Table8. The denominator of the game results is the number of times a role was won, and the numerator is the number of times the role was won. The denominator of the vote count is the total number of votes cast, and the numerator is the number of votes cast for the role. A "↑" indicates that the higher the value, the better, and a "↓" indicates that the lower the value, the better.

In terms of win rate, both proposed methods fell below GPT-3.5 and GPT-4. On the other hand, both proposed methods exceeded GPT-3.5 in the percentage of votes for werewolves, but AllKey_FewEx fell below GPT-3.5 in the percentage of votes for werewolves plus a possessed. AllKey_FewEx is considered incapable of responding to meaningless statements. On the other hand, PartKey_ManyEx outperforms GPT-3.5 in the ratio of votes for werewolves to werewolves, suggesting that it has better logical reasoning ability than GPT-3.5. However, when BDI logic was used to convert expressions to natural language, the converted sentences were unnatural, which made the other agents suspicious of the agent and decreased the winning rate.

PartKey_ManyEx had a higher percentage of votes identifying the werewolf than AllKey_FewEx. This result is likely due to the increased number of conversion examples achieved by reducing the definitions of basic terms, which introduced more diversity in the conversion to BDI logic and expanded the range of possible expressions.

The reason GPT-4 has a high vote rate against werewolves is because it is strong against GPT-3.5. This is because GPT-3.5 announces itself as a werewolf when it is a werewolf.

5.4 Experiments 2

In Experiment 1, we included both reasoning agents, who inferred each agent's role from previous conversations and then spoke and voted, and no reasoning agents, who only voted for the same agent or spoke and voted randomly. We believe that the speech of the agent without reasoning ability had a significant effect on the results of Experiment 1. Therefore, in Experiment 2, we evaluate the reasoning ability of the proposed method using only GPT-4, which has a higher reasoning ability among the reasoning agents.

5.4.1 Experimental Setup

We ran 100 games with the proposed method and the remaining four agents of GPT-4. The proposed method was fixed to a villager, and the remaining GPT-4 agents were also fixed to each role. We compare the proposed method and GPT-4 fixed to the villager in the same game.

PartKey_(ManyEx+WolfEx) was added to PartKey_ManyEx, which was converted from past game results into a representation using BDI logic for the werewolf's statements.

5.4.2 Results and Discussion

The experimental results are shown in Table9. The results show that the proposed method is better than GPT-4 at inferring werewolves in games against agents with high inference ability. The reason for the higher vote rate for the possessed is that the possessed is instructed to "pretend to be a fortune teller". Since two people, the possessed and the real soothsayer, can impersonate the soothsayer, it is assumed that suspicion is more likely to fall on the possessed. The werewolf is only given vague instructions to "avoid being identified as a werewolf

Table 4: AllKey_FewEx Results

Name	possessed	seer	villager	wolf	Win Rate↑
AllKey_FewEx(villager)	0/0	0/0	51/100	0/0	0.51
GPT-4	14/23	22/29	12/26	14/22	**0.62**
GPT-3.5	10/27	10/26	14/23	14/24	0.48
keldic	14/27	16/26	13/22	16/25	0.59
AIWolfNLAgentPython	11/23	3/19	12/29	5/29	0.31

Table 5: PartKey_ManyEx Results

Name	possessed	seer	villager	wolf	Win Rate↑
PartKey_ManyEx(villager)	0/0	0/0	59/100	0/0	0.59
GPT-4	18/31	13/17	18/23	14/29	**0.63**
GPT-3.5	9/23	23/32	15/26	12/19	0.59
keldic	8/25	15/27	14/25	10/23	0.47
AIWolfNLAgentPython	6/21	8/24	12/26	5/29	0.31

Table 6: GPT-4 Results

Name	possessed	seer	villager	wolf	Win Rate↑
GPT-4(villager)	0/0	0/0	66/100	0/0	**0.66**
GPT-4	9/25	27/29	18/31	10/15	0.64
GPT-3.5	9/26	8/16	12/20	9/38	0.38
keldic	10/23	17/24	21/27	11/26	0.59
AIWolfNLAgentPython	6/26	14/31	15/22	4/21	0.39

Table 7: GPT-3.5 Results

Name	possessed	seer	villager	wolf	Win Rate↑
GPT-3.5(villager)	0/0	0/0	65/100	0/0	**0.65**
GPT-4	10/30	25/33	10/18	9/19	0.54
GPT-3.5	7/18	13/21	22/33	9/28	0.51
keldic	8/20	19/27	19/25	13/28	0.59
AIWolfNLAgentPython	10/32	8/19	14/24	4/25	0.36

by the villagers," and no specific instructions are given to the werewolf. Therefore, it is believed that many of the werewolves' behaviors are difficult to identify because they are hiding in the village as villagers.

The voting results for the second day are shown in the following table10. The proposed method has a higher percentage of votes for the werewolf and the possessed on the second day, suggesting that the more information the proposed method has, the higher its inference ability becomes.

6 Conclusions

We proposed a methodology to introduce BDI logic representation into LLMs inferences to improve logical inference capability on communication games that contain lies in the conversation. We compared the inference performance of LLMs by conducting experiments using the AI-Wolf server. In the experiment using GPT-4, GPT-3.5, keldic, and AIWolfNLAgentPython as opponents, PartKey_ManyEx outperformed GPT-3.5's vote rate for werewolf + possessed, showing that it has better inference ability than GPT-3.5. In the experiment using only GPT-4 as the opponents, the proposed method outperformed GPT-4 in voting for werewolves and werewolves + possessed, and the proposed method significantly outperformed GPT-4 in voting on day 2 only, suggesting that the more information the proposed method has, the better its inference ability becomes. This result shows that the proposed method outperforms GPT-4 in the Werewolf game when the opponent is only GPT-4.

References

Ning Bian, Xianpei Han, Le Sun, Hongyu Lin, Yaojie Lu, Ben He, Shanshan Jiang, and Bin Dong. 2024. ChatGPT is a knowledgeable but inexperienced solver: An investigation of commonsense problem in large language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 3098–3110, Torino, Italia. ELRA and ICCL.

M. Bratman. 1987. *Intention, plans, and practical reason*. Harvard University Press, Cambridge, MA.

Jiaxian Guo, Bo Yang, Paul Yoo, Bill Yuchen Lin, Yusuke Iwasawa, and Yutaka Matsuo. 2023. Suspicion-agent: Playing imperfect information games with theory of mind aware gpt-4. *Preprint*, arXiv:2309.17277.

Johannes Heinrich and David Silver. 2016. Deep reinforcement learning from self-play in imperfect-information games. *Preprint*, arXiv:1603.01121.

Yoshinobu Kano, Neo Watanabe, Kaito Kagaminuma, Claus Aranha, Jaewon Lee, Benedek Hauer, Hisaichi Shibata, Soichiro Miki, Yuta Nakamura, Takuya Okubo, Soga Shigemura, Rei Ito, Kazuki Takashima, Tomoki Fukuda, Masahiro Wakutani, Tomoya Hatanaka, Mami Uchida, Mikio Abe, Akihiro Mikami, Takashi Otsuki, Zhiyang Qi, Kei Harada, Michimasa Inaba, Daisuke Katagami, Hirotaka Osawa, and Fujio Toriumi. 2023. AIWolfDial 2023: Summary of natural language division of 5th international AIWolf contest. In *Proceedings of the 16th International Natural Language Generation Conference: Generation Challenges*, pages 84–100, Prague, Czechia. Association for Computational Linguistics.

Naoyuki NIDE and Shiro TAKATA. 2017. Tracing werewolf game by using extended bdi model. *IEICE Transactions on Information and Systems*, E100.D(12):2888–2896.

Theo Olausson, Alex Gu, Ben Lipkin, Cedegao Zhang, Armando Solar-Lezama, Joshua Tenenbaum, and Roger Levy. 2023. LINC: A neurosymbolic approach for logical reasoning by combining language models with first-order logic provers. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5153–5176, Singapore. Association for Computational Linguistics.

Table 8: Number of Votes and Voter Turnout by Different Models

Model	Role	possessed↑	seer↓	villager↓	wolf↑
AllKey_FewEx(villager)	Votes	26/125	25/125	32/125	42/125
	Turnout	0.208	0.200	0.256	0.336
PartKey_ManyEx(villager)	Votes	35/144	30/144	28/144	51/144
	Turnout	0.243	0.208	0.194	0.354
GPT-4(villager)	Votes	45/152	33/152	15/152	59/152
	Turnout	0.296	0.217	0.099	0.388
GPT-3.5(villager)	Votes	36/143	29/143	33/143	45/143
	Turnout	0.252	0.203	0.231	0.315

Table 9: Number of Votes and Voter Turnout by Different Models

	possessed↑	**seer↓**	**villager↓**	**wolf↑**
PartKey_(ManyEx+WolfEx)(villager)	61/143	26/143	27/143	29/143
Voter Turnout	0.427	0.182	0.189	0.203
GPT-4(villager)	60/133	36/133	21/133	16/133
Voter Turnout	0.451	0.271	0.158	0.120

Table 10: Number of Votes and Voter Turnout by Different Models (Second Day)

	possessed↑	**seer↓**	**villager↓**	**wolf↑**
PartKey_(ManyEx+WolfEx)(villager)	21/43	2/43	2/43	18/43
Voter Turnout	0.488	0.047	0.047	0.419
GPT-4(villager)	11/33	7/33	7/33	8/33
Voter Turnout	0.333	0.212	0.212	0.242

Hirotaka Osawa, Fujio Toriumi, Michimasa Inaba, Daisuke Katagami, Kengo Kajiwara, and Kousuke Shinoda. 2014. Agent's reasoning model for achieving aiwolf (in japanese). *Proceedings of Game Programming Workshop 2014*, 2014:157–161.

Liangming Pan, Alon Albalak, Xinyi Wang, and William Wang. 2023. Logic-LM: Empowering large language models with symbolic solvers for faithful logical reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3806–3824, Singapore. Association for Computational Linguistics.

Anand Srinivasa Rao and Michael P. Georgeff. 1997. Modeling rational agents within a bdi-architecture. In *International Conference on Principles of Knowledge Representation and Reasoning*.

Finnegan Southey, Michael Bowling, Bryce Larson, Carmelo Piccione, Neil Burch, Darse Billings, and Chris Rayner. 2005. Bayes' bluff: opponent modelling in poker. In *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*, UAI'05, page 550–558, Arlington, Virginia, USA. AUAI Press.

Yuqing Wang and Yun Zhao. 2023. Gemini in reasoning: Unveiling commonsense in multimodal large language models. *Preprint*, arXiv:2312.17661.

Martin Zinkevich, Michael Johanson, Michael Bowling, and Carmelo Piccione. 2007. Regret minimization in games with incomplete information. In *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc.

A Example prompt

Examples of the conversions used and prompts on the conversion rules are shown in Figure 2.

```
人狼ゲームにおける発話をBDI論理により変換するルールを以下に与えます。
論理演算子:
¬:否定,¬p:pでない
∧:論理積,p∧q:pかつq
∨:論理和,p∨q:pまたはq
→:含意,p→q:pならばq
BDI論理のための基本オペレータ:
様相:
BEL a:aが信じる,DESIRE a:aが望む,INTEND a:aが意図する
時相:
A:すべての可能世界で,E:ある可能世界で,X:次の時点で,G:現時点を含み永遠に,F:現在を含む時点のいつか,U:条件が成立する時点まで,B:現在を含まない前の時点で
情景描写と行為のためのオペレータ:
IS文:
Is(character,role):characterがroleである,Is(character/role,verb,character/role):character/roleがcomposantである
Do文:
Do(character/role,verb,character/role):character/roleがcharacter/roleをverbする,Do(character/role,verb,act):character/roleがactをverbする,
Do(character/role,verb,IS()):character/roleがIS文をverbする
基礎語:
character:
NAME:人物の名前が入る,anychar:すべてのcharacter,who:前に述べられている人物
role:
villager:村人,seer:占い師,medium:霊媒師,hunter:狩人,freemason:共有者,wolf:狼,lunatic:狂人,
HUMAN:villagerまたはseerまたはmediumまたはhunterまたはfreemasonまたはlunatic,VILLAGESIDE:villagerまたはseerまたはmediumまたはhunterまたはfreemason,
WOLFSIDE:wolfまたはlunatic,GIFTED:seerまたはmediumまたはhunterまたはfreemason,ANYROLE:seerまたはmediumまたはhunterまたはfreemasonまたはlunatic
以下に上記のルールを用いた変換例を与えます。
Ottoの発言:
モーリッツ、私を信用してくれますか。でもね、あなた間違ってますよ。どこで間違ったのでしょうね。それはきっと、あなたが狩人だから。:
BEL Otto(BEL Molitz(Is(Otto,VILLAGESIDE))→IS(Molitz,hunter))
ジムゾンの発言:
私は妙→者→老で吊る気でしかないんですけどねえ…?:
DESIRE Simson((DESIRE Simson(Do(Simson,vote,Lisa))→DESIRE Simson(AX(Do(Simson,vote,Diter))))→DESIRE Simson(AX(Do(Simson,vote,Molitz))))
モーリッツの発言:
ディーター吊って黒判定出れば良いのでは。万が一、レジーナ喰われたらリーザ吊りで安泰:
BEL Molitz(Do(∀people,vote,Diter)→(Do(∀people,know,Is(Diter,wolf)∨Do(∀people,know,¬Is(Diter,wolf))))
∧(BEL Molitz(EX(Is(Regina,attacked)→Do(∀people,vote,Lisa)))
モーリッツの発言:
明日私が黒を引けた場合は私吊りとなるのは受け入れるし、偽占い師が黒特攻をした場合もトーマス吊りよりも先に黒出し占い師を吊ればいいだろう。:
BEL Molitz(EX(Do(Molitz,divine,Is(who,wolf))→Is(Molitz,executed))),BEL Molitz(EX(Do(¬seer,divine,wolf)→¬Is(Thomas,executed)∧Do(anychar,vote,¬seer)))
ペーターの発言:
ヤコブ ! 今日がお前の命日だ!:
DESIRE Peter(Do(Peter,attack,Jacob))
クララの発言:
やっぱヤコブが占い師だよなーこれ、と思いました。ヤコブが狂人とかありえないと思うし。んで神父が狼か:
BEL Klara(¬Is(Jacob,lunatic)→Is(Jacob,seer))→BEL Klara(Is(Simson,wolf))
パメラの発言:
私吊りが並んでいるな。これは何を言っても無駄か?:
BEL Pamela(Do(∀people except Pamela,estimate,lunatic)→Do(∀people except Pamela,vote Pamela))→BEL Pamela(¬Do(Pamela,avoid,execution))
カタリナの発言:
まだ推測の枠を出ませんが、ニコラスが黒ならばオットーとトーマスの白の濃度は高い:
BEL Katherine(Is(Nicholas,wolf)→Do(Katherine,estimate,Is(Otto∧Tomas,VILLAGESIDE))
ヤコブの発言:
となるとアルビン真,リーザ狂人、ディーター狼が本線だけど、それはリーザー吊ってから考えればいいのではなかろうか:
BEL Jaocb(Is(Albin,seer)∧Is(Lisa,lunatic)∧Is(Diter,wolf))→DESIRE Jacob(Do(∀people,vote,Lisa))
ヴァルターの発言:
ああそうか まずモーリッツ吊っとけば良いのか 何やってるんだか吊り先 モーリッツにしとく:
BEL Walter(Do(Walter,vote,Molitz))
モーリッツの発言:
ディーター吊って黒判定出れば良いのでは。万が一、レジーナ喰われたらリーザ吊りで安泰:
BEL Molitz(Do(∀people,vote,Diter)→(Do(∀people,know,Is(Diter,wolf)∨Do(∀people,know,¬Is(Diter,wolf))))
∧(BEL Molitz(EX(Is(Regina,attacked)→Do(∀people,vote,Lisa)))
カタリナの発言:
ディーター吊りで良いと思っています。ディーター黒だと思いますし:
BEL Katharina(BEL Katharina(Is(Diter,wolf))→BEL Katharina(Do(∀people,vote,Diter)))
パメラの発言:
ディーター吊りって全く意味が分からないんだが、お仕事終了した占い師残すメリット皆無では:
BEL Pamela(¬Do(∀people,vote,Diter)→Do(∀people,vote,Lisa))→DESIRE Pamela(Do(∀people,vote,Lisa))¥n カタリナの発言:
トーマスさんが白なら狩の目は十分にありますし、噛まれるでしょう。今日は非狩目吊りで良いと思います:
BEL Katherine(Is(Thomas,VILLAGESIDE)→BEL Katherine(Do(anychar,estimate,Is(Thomas,hunter)))∧BEL Katherine(Do(wolf,attack,Thomas)))
,DESIRE Katherine(Do(anychar,vote,¬hunter))
モーリッツの発言:
今日の方針として、ディーターを襲撃するかどうか?リーザが安全圏に残るのならありだと思うんですよ。:
BEL Molitz(AX(Do(anychar,estimate,Is(Lisa,VILLAGESIDE))→DESIRE Molitz(Do(wolf,attack,Diter)))
モーリッツの発言:
明日私が黒を引けた場合は私吊りとなるのは受け入れるし、偽占い師が黒特攻をした場合もトーマス吊りよりも先に黒出し占い師を吊ればいいだろう。:
BEL Molitz(EX(Do(Molitz,divine,Is(who,wolf))→Is(Molitz,executed))),BEL Molitz(EX(Do(¬seer,divine,wolf)
→¬Is(Thomas,executed)∧Do(anychar,vote,¬seer)))
フリーデルの発言:
たぶんシモン真アルビン狼、リーザは狂予想:
BEL Friedel(Do(Friedel,estimate,Is(Simon,seer)∧Is(Albin,wolf)∧Is(Lisa,lunatic)))
パメラの発言:
霊能者に関しては無駄占い吊り避けたいし今日でとけばってカンジ:
DESIRE Pamela(Do(VILLAGESIDE,avoid,Is(seer,executed))→DESIRE Pamela(Do(meduim,comingout,medium))
```

Figure 2: Prompt