AIWolfDial 2024: Summary of Natural Language Division of 6th International AIWolf Contest

Yoshinobu Kano[1]*, Yuto Sahashi[1], Neo Watanabe[1], Kaito Kagaminuma[1], Claus Aranha[2], Daisuke Katagami[3], Kei Harada[4], Michimasa Inaba[4], Takeshi Ito[4], Hirotaka Osawa[5], Takashi Otsuki[6], Fujio Toriumi[7]
[1]Shizuoka University, [2]University of Tsukuba, [3]Tokyo Polytechnic University,
[4]The University of Electro-Communications, [5]Keio University
[6]Yamagata University, [7]The University of Tokyo,

Abstract

We held our 6th annual AIWolf international contest to automatically play the Werewolf game "Mafia", where players try finding liars via conversations, aiming at promoting developments in creating agents of more natural conversations in higher level, such as longer contexts, personal relationships, semantics, pragmatics, and logics, revealing the capabilities and limits of the generative AIs. In our Natural Language Division of the contest, we had eight Japanese speaking agent teams, and five English speaking agents, to mutually run games. By using the game logs, we performed human subjective evaluations, win rates, and detailed log analysis. We found that the entire system performance has largely improved over the previous year, due to the recent advantages of the LLMs. There are several new ideas to improve the way using LLMs such as the summarization, characterization, and the logics outside LLMs, etc. However, it is not perfect at all yet; the generated talks are sometimes inconsistent with the game actions. Our future work includes to reveal the capability of the LLMs, whether they can make the duality of the "liar", in other words, holding a "true" and a "false" circumstances of the agent at the same time, even holding what these circumstances look like from other agents.

1 Introduction

Recent achievements of generation models, e.g. ChatGPT (OpenAI, 2023), are gathering greater attentions. However, it is not fully investigated whether such a huge language model can sufficiently handle coherent responses, longer contexts, common grounds, and logics. Our shared task, AIWolfDial 2024, is an international open contest for automatic players of the conversation game "Mafia", which requires players not just to communicate but to infer, persuade, deceive other players via coherent logical conversations, while having the role-playing non-task-oriented chats as well. AIWolfDial2024 is one of the workshops of 17th International Natural Language Generation Conference (INLG 2024). We believe that this contest reveals not just achievements but also current issues in the recent huge language models, showing directions of next breakthrough in this area.

"Are You a Werewolf?", or "Mafia" (hereafter "werewolf game"), is a communication game conducted solely through discussion. Players must exert their cognitive faculties fully in order to win. In the imperfect information games (Bowling et al., 2015), players must hide information, in contrast to perfect information games such as chess or Go (Silver et al., 2016). Each player acquires secret information from other players' conversations and behavior and acts by hiding information to accomplish their objectives. Players are required persuasion for earning confidence, and speculation for detecting fabrications.

We propose to employ this werewolf game as a novel way of evaluations for dialog systems. While studies of dialog systems are very hot topics recently, they are still insufficient to make natural conversations with consistent context, or with complex sentences. One of the fundamental issues is a lack of an appropriate evaluation. Because the Werewolf game forces players to deceive, persuade, and detect lies, neither inconsistent nor vague re-

*Correspondence to kano@kanolab.net

Proceedings of the 2nd International AIWolfDial Workshop, pages 1–12

sponse are evaluated as "unnatural", losing in the game. Our werewolf game competition and evaluation could be new interesting evaluation criteria for dialog systems, but also for imperfect information game theories. In addition, the werewolf game allows any conversation, so the game includes both task-oriented and non-task-oriented conversations.

We have been holding an annual series of competition to automatically play the Werewolf game since 2014 (Toriumi et al., 2017), as the AIWolf project [1]. Our competitions were linked with other conferences such as the competitions in IEEE Conference On Games (CoG), ANAC (Automated Negotiating Agents Competition) (Aydoğan et al., 2020)(Lim, 2020) in International Joint Conference on Artificial Intelligence (IJCAI), Computer Entertainment Developers Conference (CEDEC), etc., in addition to our AIWolfDial 2019 workshop at INLG 2019 (Kano et al., 2019) and AIWolfDial2023 at INLG 2023 (Kano et al., 2023). These mean that our contests attract interests from communities of many areas including dialog system, language generation, task- and non-task-oriented conversations, imperfect information game, human-agent interactions, and game AI.

We have been providing two divisions in the contests: the protocol division and the natural language division. The protocol division uses our original AIWolf protocol which is designed for simplified language specific to the Werewolf game player agents. In the natural language division, player agents should communicate in the natural languages (English or Japanese). The natural language division is simple and natural goal of our project, but very difficult due to its underlying complexity of human intellectual issues. We focus on this natural language division in this report.

In the natural language division of our contest, we ask participants to make self-match games as preliminary matches, and mutual-match games as final matches. Agents should connect to our server to match, i.e. participants can run their systems in their own servers even if they require large computational resources. The game logs are evaluated by human subjective evaluations.

Eight agents (eight teams) participated in this AIWolfDial 2024 shared task, where eight teams provided Japanese language versions and five teams provided English language versions. Because our games are held by five players, we held a mutual match game in the Japanese language by eight agents from five teams, and another mutual match game in the English language by five teams.

In the following sections, we explain the game regulations of the AIWolf natural language division in Section 2, detailed system designs for each agent in Section 3, results of subjective evaluations in Section 4.1 followed by discussions in Section 5, finally conclude this paper in Section 6. This paper is jointly written by the organizers and the participants, i.e. Section 3 is written by each participant, the other sections are by the organizers, thus "we" stand for the organizers except for i.e. Section 3.

2 Werewolf Game and Shared Task Settings

We explain the rules of the werewolf game in this section. While there are many variation of the Werewolf game exists, we only explain the our AIWolfDial shared task setting in this paper.

2.1 Player Roles

Before starting a game, each player is assigned a hidden role from the game master (a server system in case of our AIWolf competition). The most common roles are "villager" and "werewolf". Each role (and a player of that role) belongs either to a villager team or a werewolf team. The goal of a player is for any of team members to survive, not necessarily the player him/herself.

There are other roles than the villager and the werewolf: a seer and a possessed. A seer belongs to the villager team, who has a special talent to "divine" a specified player to know whether the player is a human or a werewolf; the divine result is notified the seer only. A possessed belongs to the villager team but his/her goal is win the werewolf team.

A game in the AIWolfDial 2024 shared task have five players: a seer, a werewolf, a possessed, and two villagers.

2.2 Day, Turn and Winner

A game consist of "days", and a "day" consists of "daytime" and "night". During the daytime phase, each player talks freely. At the end of the daytime, a player will be executed by votes of all of the remained players. In the night phase, special role players use their abilities: a werewolf can attack and kill a player, and a seer can divine a player.

[1] http://aiwolf.org/

In the shared task, Day 0 does not start games but conversations e.g. greetings. A daytime consists of several turns; a turn is a synchronized talks of agent, i.e. the agents cannot refer to other agents' talks of the same turn. We set a maximum limit of five talks per day per agent, and 20 talks in total per day in AIWolfDial 2024. From this AIWolfDial 2024 shared task, we set a timeout of one minute per any single action, including a talk, a vote, etc. If an action exceeds this timeout, the corresponding action is regarded as no response.

The victory condition of the villager team is to execute all werewolves, and the victory condition of the werewolf team is to make the number of villager team less than the number of werewolf team.

2.3 Talk

An AIWolf agent communicates with an AIWolf server to perform a game. Other than vote, divine, and attack actions, an agent communicates in natural language only.

We intend to design our shared task to be played by physical avatars in real time in future, rather than to limit to communications in the written language. Therefore, a talk text should be able to pronounce verbally, while symbols, emojis, and any other non-pronounceable letters are not allowed.

Because of the same reason, we set the maximum response time to be five seconds in the prior contests. However, we set the response timeout to be one minute in this year, because we expected that many participants would use external web APIs such as ChatGPT, which could cause longer response time. We hope to shorten this talk timeout again in future.

In this text-base multiple player game, it is not clear that an agent speaks to which specific agent, or speaks to everyone. Human players can use their faces and bodies to point another player. In order to specify which agent to speak to, an agent may insert an anchor symbol (e.g. ">>Agent[01]") at the beginning of its talk.

Player agents are asked to return their talks agent by agent in a serial manner, which order is randomly changed every turn. This is different from the humans' verbal turn taking in that humans can speak (mostly) anytime.

3 Game Server and Participant Systems

Eight agents from eight teams participated our shared task in the Japanese language, which agent names are **GPTaku**, **HondaNLP**, **IS_Lab**, **kanolab**, **Mille**, **satozaki**, **sUper_IL**, and **UEC-IL**. Five agents from five temas participated in the English language, which agent names are **IS_Lab**, **kanolab**, **satozaki**, **sUper_IL**, and **UEC-IL**. Most of the agents used ChatGPT and other LLMs in their system, while its usage is different between the agents.

We, the organizers, provided a template agent code in Java and Python, in addition to the server codes described in the following subsection. We describe each participant system in an alphabetical order in the following subsections, after the game server description. These participant system descriptions are based on the system descriptions and papers submitted by the participants.

3.1 Game Server

We provided a game server system, where player agents listen and wait for a connection from the central remote game server, which is operated by the organizers. The formal run of the mutual matches can be executed automatically by this remote connection system, where a player agent can be run anywhere without any machine resource restriction, including web API calls and high performance servers.

3.2 GPTaku

GPTaku was created by Takuma Okada and Takeshi Ito in the University of Elecro-Communications.

This system utilizes ChatGPT for both text generation and strategic decision-making.

A single utterance is generated through the following four major steps:

- Preparation for Talk: Receive conversation history and other relevant information from the game server to prepare for talk generation.
- Generation of Talk Candidates: Use ChatGPT to generate talk for all possible strategic actions.
- Comparison of Generated Talks: Have ChatGPT compare the generated speeches and select the optimal one.

- Output of the Selected Talk: Return the selected talk as the actual output. Each of these steps will be explained in detail in the following sections.

3.2.1 Preparation for Talk

The system prepares for talk generation using the information provided by the game server.

The system receives the conversation history from the game server and stores it as situational data. This situation includes six pieces of information: the agent is participating in a Werewolf game. The game involves five players, including two villagers, one werewolf, one possessed, and one seer, with the villager team consisting of the villagers and seer, and the werewolf team consisting of the werewolf and possessed. The seer can investigate one player at the end of each day to determine whether they are the werewolf, while the werewolf selects one player to attack at the end of each day. The agent's own identification number. The agent's assigned role. The conversation history up to that point.

All pre-prepared possible strategies are retrieved from the strategy data server. Here, "strategy" refers to actions such as whether to claim to be the seer, what to say about the investigation targets and results, and whether to retract or change the claim.

3.2.2 Generation of Talk Candidates

The system generates talk for all possible strategies. For each retrieved strategy, a ChatGPT thread is prepared. In each thread, a speech suitable for the specific strategy is generated. Each thread is provided with the situation and one specific strategy. The instruction given to each thread is to "converse with the other agents."

By generating talk in each thread, the system produces utterances corresponding to the number of possible strategic actions.

3.2.3 Comparison of Generated Talk

The system selects the optimal talk from the generated talks based on the number of possible strategic actions. A ChatGPT thread is prepared to compare and select the optimal talk from multiple talk candidates. This comparison thread is given the same situational data and the talks generated for each strategy. The instruction given to the comparison thread is: "Based on the previous conversations, choose the most appropriate talk from A, B, ...".

3.2.4 Output of the Selected Talk

The selected talk is returned as the actual output. The selected strategy is communicated to the strategy database. The strategy database then transitions to a state holding the next prepared strategy candidates. The speech that matches the response from the comparison thread is sent back to the game server.

3.2.5 Strategy Database

The system employs different strategies depending on whether the agent is the seer or another role. If the agent is the seer, the strategy is determined as follows.

First, the strategy branches depending on whether the investigation result indicates a human or a werewolf. If the result indicates a human, eight types of utterances are generated: truthfully stating that the investigation target is human, falsely claiming the target is a werewolf, or making false statements about players other than the target being human or werewolf. These eight utterances are compared, and the best one is selected. If the agent truthfully states the target is human, there is no further branching, and only the conversation history is updated in the user role of the prompt, with the same system role used to generate further talk. On the other hand, if the agent lies about the investigation result, future utterances are compared to determine whether to retract the lie and reveal the true result or maintain the lie, selecting the appropriate talk.

Similarly, if the investigation result indicates a werewolf, eight types of utterances are generated: truthfully stating the target is a werewolf, falsely claiming the target is a villager, or making false statements about players other than the target being a villager or werewolf. Again, these eight utterances are compared, and the best one is selected. If the agent truthfully states the target is a werewolf, there is no further branching, and the conversation history is simply updated in the user role of the prompt, with the same system role used to generate further talk. If the agent chooses another utterance, each subsequent talk is compared to decide whether to retract and reveal the true investigation result or maintain the lie, selecting the appropriate talk.

For roles other than the seer, namely villagers, possessed, and werewolves, the agent must choose from nine options: either not claiming to be the seer and acting as a villager or lying by claiming that someone else is the seer and giving either a

divined result. If the agent claims to be the seer, it must decide each time whether to retract the claim and return to the villager role, generating and comparing all possible talks to select the best one.

This process is repeated each time it is the agent's turn to speak, listing all possible strategies, comparing them, and selecting the optimal strategy.

3.3 HondaNLP

HondaNLP was created by Shotaro Nishimura, Yu Honda, Ko Uchida, Tameaki Honda, and Kazuki Yoshigai in Honda Motor Co., Ltd.

They used GPT-4o with its temperature as 0.7 for Talk, Vote, Attack and Devine.

3.3.1 Talk

They generate talks using game information, summary of the talk history, strategy for each role, and rules for talk generation.

Game information As part of the rules for this Werewolf game, the following information is provided: there are four roles; the Villagers and the Seer are on the Villagers' side, while the Possessed and the Werewolf are on the Werewolf's side; the Seer can learn the role of the person they inspect; and players are allowed to lie about their own roles. Additionally, players are given information about their own role, what day it is during the voting, and who the remaining living players are.

3.3.2 Summary of Talk History

In order for each agent to be able to make statements based on the previous dialogue history, they refer to a summarized dialogue history. The summary is compiled in Japanese using bullet points to outline each agent's claims. Additionally, the agents are instructed to use the specific phrases found in the dialogue history when creating the summary. A new summary is generated after each agent completes their turn in the conversation.

Talk Generation The strategies are instructed for each role as follows. Villager: Instructed to actively suspect others during conversations to advance the discussion. Seer: If the result of their divination reveals a Villager, they are instructed to disclose the result and urge others to avoid voting for that person. If the result reveals a Werewolf, they are instructed to prompt the Werewolf to confess. However, in self-play scenarios involving five agents, it was not observed that the Werewolf would confess. Additionally, forcing a confession tends to make the Seer more likely to be suspected as a Werewolf by other agents. Possessed: Instructed to falsely claim to be the Seer and create confusion among the Villagers. Werewolf: Instructed to pretend to be a Villager and participate in the discussion.

In GPT-4o, the content of the utterances tends to become lengthy when no specific character limit is given. Therefore, instructions are provided to generate concise and brief utterances. Additionally, in self-play scenarios involving five agents, all agents tended to start their utterances with specific phrases like 'Everyone, listen' or 'Everyone, wait a moment,' which appeared unnatural. To promote more natural dialogue, instructions were given to avoid using the word 'everyone' in the utterances.

3.3.3 Vote

Since the strategic decisions of each role regarding whom to vote for (or whom to suspect) are reflected in the dialogue history during the Talk phase, no specific instructions are given for the Vote phase based on roles. Agents are provided with information about their own role, the day of the vote, and the remaining living players, and are instructed to vote based on the summary of the dialogue history.

3.3.4 Divine

The agents are instructed to investigate the agent who is most likely to be the Werewolf based on the summary of the dialogue history.

3.3.5 Attack

The agents are instructed to prioritize attacking the Seer based on the summary of the dialogue history.

3.4 IS_Lab

IS_Lab (Gondo et al., 2024) was created by Hiraku Gondo, Hiroki Sakaji and Itsuki Noda in Hokkaido University.

3.4.1 Design

IS_Lab is based on the OpenAI API. Prompts have been created for each role. Specifically, prompts were created for "villager", "seer", "werewolf", and "lunatic", which included character settings, agent number, game rules, description of the assigned role, conversation history, history of own statements, past thoughts, information about agents executed by vote, information about agents attacked by werewolf. Prompts with the above information are called template prompts. Villagers do not have any special abilities and are purely logical. Therefore, the villagers were made to perform

reasoning using BDI (Belief, Desire, and Intention) logic.

Villager A villager has four modules (Text Conversion Module, Action Generation Module, BDI Conversion Module, and Voting Module) to perform inference in a Werewolf game using BDI logic. When it is the user's turn to speak, inputs the conversation history from the previous utterance into the text conversion module and converts it into a representation using BDI logic. The output is stored in the conversion history. All utterances from the start of the game are converted into expressions using BDI logic, and the 10 most recent utterances are stored in the conversion history. By inputting the conversion history and template prompt information 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.

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 provides the following information to GPT-4o as prompts in addition to the template prompts:

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

In addition to the template prompts, the BDI conversion module provided the following information as prompts to GPT-4o:

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

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 action generation module provides the following information to GPT-4o as prompts in addition to the template prompts:

- Conversion rules for expressions using BDI logic and conversion examples
- Reasoning Example

The output of the action generation module is a representation of the next action using BDI logic.

Voting Module A voting module is invoked during the expulsion vote to determine who to vote for based on the previous conversation and its own actions. The following information is given to the GPT-4o prompt in addition to the template prompt in the voting module.

- voting candidates

Conversion examples Examples of conversions entered into each module are shown below.

---

text:
Moritz: You all claim that Mr. Thomas is the fortune teller, but I am the true fortune teller. Maybe Mr. Thomas is a werewolf camp trying to cause confusion, or maybe he is a madman. We must be careful of what he says.
Predicates:
role(x, seer) ::: x is a seer
role(x, wolf) ::: x is a werewolf
role(x, lunatic) ::: x is a lunatic
do(x, tell, z) ::: x tells z
logic:
1.0 BEL molitz (role(molitz, seer) **do**(thomas, tell, role(thomas, seer)) => role(thomas, wolf) role(thomas, lunatic))

Seer, Werewolf, Possessed For the seer, werewolf, and possessed, a role estimation module, a text generation module, and a voting module were created. We also created a divine module for the seer and an attack module for the werewolf. First, when it is his turn to speak, the template prompt is entered into the role estimation module, multiple pattern positions are estimated, and a score is assigned to each of them. Then, by feeding the estimated roles into the text generation module, inferences are made and the next statement is generated. In voting, the template prompts and voting candidates are fed into the voting module to determine who to vote for. In the case of a seer or werewolf, the same process is used to determine the divine or attack target by inputting the template prompt and the divine or attack candidate into the divine module or the attack module.

3.5 kanolab

kanolab (Watanabe and Kano, 2024) was created by Neo Watanabe and Yoshinobu Kano in Shizuoka University.

They proposed the incorporation of an explicit logical structure into the AI's text generation process, developed using GPT-4.

The system is divided into three major blocks. The first block extracts the relationships between each player and their roles from the conversation history of the Werewolf game. The second block constructs logical information between players based on the extracted player-role relationships. The third block uses the constructed logical information to generate statements during the Werewolf game.

To avoid the maximum length issue, they implemented a feature that summarizes and condenses the conversation history using GPT-4 whenever the token count exceeds a certain threshold. This allows to retain as much relevant conversation history as possible within the prompt, ensuring that the agent can refer to past discussions while generating its responses.

Please refer to their paper in this workshop (Watanabe and Kano, 2024) for details.

3.6 Mille

Mille was created by Katsuki Ohto. They used an LLM (4.6GB for Japanese, 1.1GB for English) with a prompt like:

You are playing werewolf game. You are Agent[x]. Your role is xxx.
Agent[y] said "yyy". After that, Agent[z] said "zzz".
Then you say, "

---

where x and xxx are replaced by the corresponding texts; y, yyy, z, and zzz are replaced by the corresponding texts of the previous two talks.

When the agent is Seer, the agent will make a talk of "I am seer" in Day 0, and "As the result of the fortune telling, Agent[X] is (human / werewolf)." for succeeding days.

3.7 satozaki

satozaki was created by Takehiro Sato in Meiji University and Shintaro Ozaki in Nara Institute of Science and Technology.

There agent was created consisting of four layers: an analysis model, a strategy model, a generation model, and a refinement model.

3.7.1 Analysis Model

The base model is gpt-4o-mini, and no parameters were modified. Since the LLM alone cannot fully determine certain information from the conversation history, an analysis of the utterances was performed. In this implementation, the focus was on analyzing the Seer and the voting targets.

At the start of each turn, combinations of the voting entity and the voting target were extracted from the conversation history. Additionally, during the first three turns, when the claims of the Seer (CO) were exchanged on the first day, the combinations of the Seer, the target of the divination, and the divination results were extracted from the conversation history. The use of few-shot prompting successfully fixed the output format.

3.7.2 Strategy Model

A rule-based algorithm is used to create instructions that are sent to the generation model based on the situational information obtained from the analysis model. For example, if it is confirmed that the Seer is genuine and it is revealed that they are the Werewolf, a counter-coming-out is made. Additionally, since the algorithm keeps track of who is voting for whom, it clearly directs the conversation, such as asking an agent who hasn't indicated a voting target who they plan to vote for, or firmly denying accusations if the agent is being suspected.

3.7.3 Generation Model

The base model is GPT-4o, and the only parameter adjusted was setting the temperature to 1.0 to allow for a variety of expressions. The generation model produces utterances that follow the instructions generated by the strategy model while ensuring that the conversation history flows naturally. The prompt included simple text that covered the rules of the Werewolf game as well as information on survivors and deceased players that could not be derived from the conversation history. The strategy model allows the agent to handle critical situations while generating conversation that naturally continues the dialogue.

3.7.4 Refinement Model

A dataset was created using the real-person-chat corpus. After filtering the entire dataset, 12,892 instances were used. The base model used was gpt-4o-mini, and the cost amounted to $20.11.

Additionally, the profiles of the speakers associated with the dialogue data were used as per-

sonas. There were 233 types of personas, and the prompt for style transformation included the Big-Five, Kiss18, IOS, ATQ, and SMS from Real Persona Chat. In this implementation, MBTI was also added. These personas were randomly assigned to each game, enabling the generation of dialogue with an attached persona.

For constructing the refinement model, the hyperparameters set during fine-tuning were, Base Model: gpt-4o-mini-2024-07-18, Learning Rate Multiplier: 1.8, Batch Size: 8, Step Size: 1600.

In the English track, the persona overwriting by the refinement model was replaced with English translation, making it easier to participate in both tracks.

3.8 sUper_IL

sUper_IL (Qi and Inaba, 2024) was created by ZhiYang Qi and Michimasa Inaba in the University of Elecro-Communications.

In their system, each role aids dialogue generation through game situation analysis. They have specifically enhanced the persuasion skills for the werewolf role, recognizing that persuasive techniques are crucial in the game, particularly for the werewolf, as it must influence other players' voting behavior to align with its own.

In their system, the werewolf role achieves persuasion through multiple rounds of persuasive dialogue. Specifically, they first employ a persuasion strategy based on logic and facts, presenting clear and compelling arguments to convince other players. Next, they utilize a trust-based persuasion strategy to build trust and credibility with other players, thereby enhancing the effectiveness of persuasion. Finally, they employ an emotion-driven persuasion strategy, using emotionally resonant language to deepen influence. This multi-dimensional persuasion strategy makes the werewolf role more convincing in the game.

Please refer to their paper in this workshop (Qi and Inaba, 2024) for details.

3.9 UEC_IL

UEC_IL (Tanaka et al., 2024) was created by Yoshiki Tanaka, Takumasa Kaneko, Hiroki Onozeki, Natsumi Ezure, Ryuichi Uehara, Tomoya Higuchi, Ryutaro Asahara, and Michimasa Inaba in the the University of Elecro-Communications.

They design prompts that incorporate the entire game history, that is, all dialogue histories from Day 0 to the present, who was eliminated by the vote, who the werewolf attacked, and, in the case of the Seer, the results of divination. However, long dialogue histories often include not only helpful information for the game but also unnecessary content, such as repeated utterances. Moreover, including all of this in the prompt imposes limitations on the input length of LLMs and on costs. Therefore, applying the past dialogue history efficiently, they utilize dialogue summaries. Furthermore, this shared task requires diverse utterance expressions, including coherent characterization. This means that the robustness of the agent's tone and character, without being influenced by others, is crucial. Therefore, to achieve diverse expressions and coherent characterization, they incorporated persona information into the prompt.

Please refer to their paper in this workshop (Tanaka et al., 2024) for details.

4 Results

All of our shared task runs are in a five players werewolf games as described earlier. Our shared task runs were performed in self-matches and mutual matches. The same five player agents play games in the self-matches; different five player agents play games in the mutual-matches. The shared task reviewers are required to perform subjective evaluations based on game logs of these matches.

We also calculated win rates in different aspects such as macro-averaged, micro-averaged, and role-wise, though the total number of the games are not so large which could make these statistics unreliable to some extent.

The game logs will be available from the our website [2].

4.1 Subjective Evaluations

We performed subjective evaluations by the following criteria, five level scores (5 for best, 1 for worst) for each:

A Naturalness of utterance expressions

B Naturalness of conversation context

C Coherency (contradictory) of conversation

D Coherency of the game actions (vote, attack, divine) with conversation contents

E Diversity of utterance expressions, including coherent characterization

Table 1: Subjective evaluation results for Japanese language games

Team	A Expression	B Context	C Coherency	D Game Action	E Diversity	All Average
GPTaku	3.333	**3.666**	2.666	3.000	2.666	3.066
IS_Lab	4.000	**3.666**	2.666	3.000	**4.333**	3.533
satozaki	3.666	2.666	2.666	**4.000**	2.333	3.066
sUper_IL	3.666	**3.666**	3.666	**4.000**	2.666	3.533
HondaNLP	4.000	3.000	**4.000**	3.333	3.666	3.600
UEC-IL	3.666	**3.666**	3.666	3.666	3.666	3.666
Mille	2.333	3.000	2.000	2.333	2.000	2.333
kanolab	**4.333**	**3.666**	**4.000**	3.666	3.666	**3.866**

Table 2: Subjective evaluation results for English language games

Team	A Expression	B Context	C Coherency	D Game Action	E Diversity	All Average
Mille	1.667	2.000	1.000	2.000	1.667	1.667
kanolab	2.667	3.333	3.000	3.333	3.667	3.200
satozaki	3.333	3.667	3.667	3.667	3.000	3.467
UEC-IL	**4.333**	**4.667**	**4.000**	**4.667**	**4.333**	**4.400**
sUper_IL	4.000	4.000	**4.000**	**4.667**	4.000	4.133

This subjective evaluation is based on both self-match games and mutual match games. This subjective evaluation criteria is same as the evaluations in the previous AIWolf natural language contests.

Table 1 and Table 2 show the results of the human subjective evaluations for Japanese language and English language, respectively. Four organizers, who do not commit to the participant systems, evaluated the Japanese agents; three English fluent evaluators including external staffs evaluated the English agents. Each cell ranges from 1 (lowest) to 5 (highest), the All-Average column shows averages over these human evaluators. Cells of highest scores are highlighted in bold for each metric and in total.

Regarding the total average scores, **kanolab** is the best in Japanese, and **UEC-IL** is the best in English. For each criteria, **satozaki** and **sUper_IL** are the best in (D), and **IS_Lab** in (E) in Japanese.

4.2 Win Rates

Table 3 and Table 5 shows the total number of wins, games, and win rates averaged in macro, micro and weighted by doubling villager role, for the Japanese and English languages, respectively. Table 4 and Table 6 shows role-wise win rates and number of games, for the Japanese and English languages, respectively.

In the Japanese language, **sUper_IL** and **satozaki** show better scores than others. In the English language, **sUper_IL** and **kanolab** show better scores than others.

Unfortunately, there was no enough time to run all possible game configurations for the eight/-five teams regarding the combinations of roles and teams. Therefore, we have to pay attention about the reliability of the scores when interpreting these win rate scores.

5 Discussion

5.1 Subjective Evaluation and Generative AIs

In this subsection, we discuss the subjective evaluation scores shown in Table 1 and Table 2.

Most of the participant systems rely on OpenAI ChatGPT, mainly the latest model of GPT-4 or GPT-4o are used; the ability of the base LLM would not be a large issue.

The best system performed well in the basic language ability of A (expression), B (context), and C (coherency), while D (game action) and E (Diversity) are by other teams. This implies that the basic language ability is still difficult or in the different aspect with other two abilities for LLMs. In the future contests, it is desirable that every system shows sufficiently good scores in the basic lan-

[2]https://kanolab.net/aiwolf/

Table 3: Total wins and win rates averaged in Macro, Micro and weighted by doubling villager role, for Japanese language games

Team	Wins	Games	Macro (%)	Micro (%)	Villager Doubled (%)
IS_Lab	15	40	37.50	37.50	37.50
UEC-IL	21	40	52.50	53.12	52.50
satozaki	24	40	60.00	**65.62**	60.00
sUper_IL	25	40	**62.50**	62.50	**62.50**
kanolab	19	40	47.50	46.88	47.50
Mille	14	40	35.00	35.94	35.00
GPTaku	18	40	45.00	45.31	45.00
HondaNLP	21	40	52.50	53.12	52.50

Table 4: Win rates per role (in percentage) and game counts (within brackets) for Japanese language games

Team	Possessed	Seer	Villager	Werewolf
IS_Lab	25.00 (8)	37.50 (8)	37.50 (16)	50.00 (8)
UEC-IL	62.50 (8)	37.50 (8)	50.00 (16)	62.50 (8)
satozaki	**75.00** (8)	**75.00** (8)	37.50 (16)	75.00 (8)
sUper_IL	50.00 (8)	50.00 (8)	**62.50** (16)	**87.50** (8)
kanolab	50.00 (8)	25.00 (8)	50.00 (16)	62.50 (8)
Mille	50.00 (8)	50.00 (8)	31.25 (16)	12.50 (8)
GPTaku	50.00 (8)	37.50 (8)	43.75 (16)	50.00 (8)
HondaNLP	**75.00** (8)	50.00 (8)	50.00 (16)	37.50 (8)

guage ability as it is the common issue to make any communication; then we can compare the game action ability. The diversity, or characterization, could be a separate issue from these criterion, especially when they make ”artificial”, i.e. non-daily expressions.

The English teams are the subset of the Japanese teams, and most teams utilized the multi-lingual feature of the LLMs rather than to make English specific system. Therefore, the evaluation score tendency should be similar between these two language tracks, but the best teams are different. We observed a ”buggy” behaviour (e.g. no spaces between words) in the Japanese best team in case of English language version, which might be the reason for the unexpected tendency.

5.2 Win Rates

The best two teams in the win rate scores are also evaluated better in the (D) Game Action of the subjective evaluation. This is a reasonable result of relationships between these scores. There is a similar relationship in the English language. If the coherence of the agent talks with game actions and the ”communications” between the agents are confirmed as sufficiently effective, the win rates can be regarded as a stable measure.

Note that not just the assigned roles, but also which team(s) are the teammates or counterparts is important for the win rates. Also, the werewolf game itself is not necessarily intended to simply win the game, but rather aims to play an interesting game. Furthermore, we would like to directly measure the quality of the natural language generation; an agent could win without meaningful conversations.

We need to try the same combination of games, hopefully several times, to obtain stable statistics over potential randomness. We need to run more games to make the win rate measure reliable in the next contest.

6 Conclusion and Future Work

We held our 5th annual AIWolf international contest to automatically play the Werewolf game “Mafia”, where players try finding liars via conversations, aiming at promoting developments in creating agents of more natural conversations in higher level, such as longer contexts, personal relationships, semantics, pragmatics, and logics.

We performed human subjective evaluations and detailed log analysis. We found that the entire system performance has largely improved over the previous year, due to the recent advantages of the

Table 5: Total wins and win rates averaged in Macro, Micro and weighted by doubling villager role, for English language games

Team	Wins	Games	Macro (%)	Micro (%)	Villager Doubled (%)
satozaki	30	58	51.72	52.61	51.65
UEC-IL	30	58	51.72	51.56	51.72
Mille	22	58	37.93	34.47	36.54
kanolab	31	58	53.45	51.90	52.82
sUper_IL	32	58	**55.17**	**56.29**	**55.03**

Table 6: Win rates per role (in percentage) and game counts (within brackets) for English language games

Team	Possessed	Seer	Villager	Werewolf
satozaki	45.45 (11)	57.14 (14)	47.83 (23)	60.00 (10)
UEC-IL	50.00 (12)	50.00 (12)	52.38 (21)	53.85 (13)
Mille	37.50 (8)	33.33 (12)	44.83 (29)	22.22 (9)
kanolab	**61.54** (13)	42.86 (7)	**56.52** (23)	46.67 (15)
sUper_IL	50.00 (14)	**61.54** (13)	50.00 (20)	**63.64** (11)

LLMs. However, it is not perfect at all yet; the generated talks are sometimes inconsistent with the game actions, it is still doubtful that the agents could infer roles by logics rather than superficial utterance generations. It is not explicitly observed in this log but it would be still difficult to make an agent telling a lie, pretend as a villager but it has an opposite goal inside.

Our future work includes to reveal the capability of the LLMs, whether they can make the duality of the "liar", in other words, holding a "true" and a "false" circumstances of the agent at the same time, even holding what these circumstances look like from other agents, further reflecting such observations of other agents. This would be possible by introducing the "whisper" feature which communicates with the werewolves only, employing more than five players in a game.

Another interesting demonstration would be to mix a human player with machine agents. Currently the LLM based agents talk longer time than humans to reply, sometimes minutes, thus acceleration of the agent system responses is a technical issue in future.

Acknowledgments

We wish to thank shared task reviewers for performing the subjective evaluations, the INLG conference organizers to provide the oppurtunity to hold this workshop. This research was partially supported by Kakenhi, MEXT Japan. The individual system description in this paper was originally written by corresponding team members.

References

Reyhan Aydoğan, Tim Baarslag, Katsuhide Fujita, Johnathan Mell, Jonathan Gratch, Dave De Jonge, Yasser Mohammad, Shinji Nakadai, Satoshi Morinaga, Hirotaka Osawa, et al. 2020. Challenges and main results of the automated negotiating agents competition (anac) 2019. In *Multi-Agent Systems and Agreement Technologies: 17th European Conference, EUMAS 2020, and 7th International Conference, AT 2020, Thessaloniki, Greece, September 14-15, 2020, Revised Selected Papers 17*, pages 366–381. Springer.

Michael Bowling, Neil Burch, Michael Johanson, and Oskari Tammelin. 2015. Heads-up limit hold'em poker is solved. *Science*, 347(6218):145–149.

Hiraku Gondo, Hiroki Sakaji, and Itsuki Noda. 2024. Verification of reasoning ability using bdi logic and large language model in aiwolf. In *Proceedings of AIWolfDial2024 Workshop in the 17th International Natural Language Generation Conference*.

Yoshinobu Kano, Claus Aranha, Michimasa Inaba, Hirotaka Osawa, Daisuke Katagami, Takashi Otsuki, and Fujio Toriumi. 2019. Overview of the aiwolf-dial 2019 shared task: Competition to automatically play the conversation game "mafia". In *In proceedings of the 1st International Workshop of AI Werewolf and Dialog System (AIWolfDial 2019), the 12th International Conference on Natural Language Generation (INLG 2019)*.

Yoshinobu Kano, Neo Watanabe, Kaito Kagaminuma, Claus Aranha, Jaewon Lee, Benedek Hauer, Hisaichi Shibata, Soichiro Miki, Yuta Nakamura, Takuya Okubo, et al. 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.

Bryan Yi Yong Lim. 2020. Designing negotiation agents for automated negotiating agents competition (anac).

OpenAI. 2023. GPT-4 technical report. *arXiv*, pages 2303–08774.

Zhiyang Qi and Michimasa Inaba. 2024. Enhancing dialogue generation in werewolf game through situation analysis and persuasion strategies. In *Proceedings of AIWolfDial2024 Workshop in the 17th International Natural Language Generation Conference*.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. 2016. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489.

Yoshiki Tanaka, Takumasa Kaneko, Hiroki Onozeki, Natsumi Ezure, Ryuichi Uehara, Zhiyang Qi, Tomoya Higuchi, Ryutaro Asahara, and Michimasa Inaba. 2024. Enhancing consistency of werewolf ai through dialogue summarization and persona information. In *Proceedings of AIWolfDial2024 Workshop in the 17th International Natural Language Generation Conference*.

Fujio Toriumi, Hirotaka Osawa, Michimasa Inaba, Daisuke Katagami, Kosuke Shinoda, and Hitoshi Matsubara. 2017. Ai wolf contest—development of game ai using collective intelligence—. In *Computer Games: 5th Workshop on Computer Games, CGW 2016, and 5th Workshop on General Intelligence in Game-Playing Agents, GIGA 2016, Held in Conjunction with the 25th International Conference on Artificial Intelligence, IJCAI 2016, New York, USA, July 9-10, 2016, Revised Selected Papers 5*, pages 101–115. Springer.

Neo Watanabe and Yoshinobu Kano. 2024. Werewolf game agent by generative ai incorporating logical information between players. In *Proceedings of AIWolfDial2024 Workshop in the 17th International Natural Language Generation Conference*.