PAPER *Special Section on Information and Communication Technology to Support Hyperconnectivity*

An Automated Multi-Phase Facilitation Agent Based on LLM

Yihan DONG[†a)], ***Nonmember***, **Shiyao DING**[†], ***and*** **Takayuki ITO**[†], ***Members***

SUMMARY This paper presents the design and implementation of an automated multi-phase facilitation agent based on LLM to realize inclusive facilitation and efficient use of a large language model (LLM) to facilitate realistic discussions. Large-scale discussion support systems have been studied and implemented very widely since they enable a lot of people to discuss remotely and within 24 hours and 7 days. Furthermore, automated facilitation artificial intelligence (AI) agents have been realized since they can efficiently facilitate large-scale discussions. For example, D-Agree is a large-scale discussion support system where an automated facilitation AI agent facilitates discussion among people. Since the current automated facilitation agent was designed following the structure of the issue-based information system (IBIS) and the IBIS-based agent has been proven that it has superior performance. However, there are several problems that need to be addressed with the IBIS-based agent. In this paper, we focus on the following three problems: 1) The IBIS-based agent was designed to only promote other participants' posts by replying to existing posts accordingly, lacking the consideration of different behaviours taken by participants with diverse characteristics, leading to a result that sometimes the discussion is not sufficient. 2) The facilitation messages generated by the IBIS-based agent were not natural enough, leading to consequences that the participants were not sufficiently promoted and the participants did not follow the flow to discuss a topic. 3) Since the IBIS-based agent is not combined with LLM, designing the control of LLM is necessary. Thus, to solve the problems mentioned above, the design of a phase-based facilitation framework is proposed in this paper. Specifically, we propose two significant designs: One is the design for a multi-phase facilitation agent created based on the framework to address problem 1); The other one is the design for the combination with LLM to address problem 2) and 3). Particularly, the language model called "GPT-3.5" is used for the combination by using corresponding APIs from OPENAI. Furthermore, we demonstrate the improvement of our facilitation agent framework by presenting the evaluations and a case study. Besides, we present the difference between our framework and LangChain which has generic features to utilize LLMs.

key words: *automated facilitation agent, consensus-making, group discussion, large language model*

1. Introduction

Consensus-building holds significant relevance in the realm of multi-agent systems, wherein the goal is to foster agreement through robust discussions. With the burgeoning growth of social networking services (SNS), these dialogues often take place in an online environment. Consequently, the allure of achieving consensus via online discussions has grown exponentially.

In tandem with this trend, there has been a surge in the design and implementation of large-scale online discussion support platforms, such as D-agree [1]. These platforms aim to facilitate more seamless and straightforward consensus-building among participants.

Within such platforms, an automated facilitation agent is integral, its primary function being to streamline and analyze conversations to expedite the consensus-building process. The current IBIS-based agent was implemented based on the argumentation-based approach called the issue-based information system (IBIS) [2]. A lot of social discussion experiments have been conducted successfully utilizing the IBIS-based agent, as seen in Refs. [3]–[6]. The effectiveness of the IBIS-based agent has been substantiated further by studies such as [7].

However, there are three major issues that hinder the further application of the IBIS-based agent: 1) The first issue is that the discussion environment is inclusive due to various characteristics of participants being ignored in the discussion, causing the consequence that the discussions are often insufficient. For example, it is far more difficult for shy participants to join in the discussion and give their own opinions than outgoing participants; 2) The second issue is that the facilitation messages generated by the IBIS-based agent were not natural enough, leading to consequences that the participants were not sufficiently promoted and the participants did not follow the flow to discuss a topic; 3) Although the above issues can be partially solved by large language models (LLM) to fully use the performance of LLM, guidance and control are necessary.

To address the aforementioned challenges and to foster and strengthen an inclusive discussion environment, we propose a large language model-based multi-phase facilitation agent. To tackle the first issue 1) and the second issue 2), we have designed and implemented functionalities such as automated facilitation for ice-breaking and promoting the discussion according to various stages. These functionalities have been integrated into the IBIS-based agent.

For the third issue 3), in order to enhance the facilitation ability of our agent, we have designed and implemented conjunction with a language learning model. This includes multiple prompts that allow the language model to not only generate appropriate messages but also to summarize the posts of participants. This feature serves to remind participants of previous ideas or arguments presented in the discussion. By comparing messages generated from template files with those generated by the language model, we can clearly illustrate the advantages of combining the automated

Manuscript received June 1, 2023.
Manuscript revised September 22, 2023.
Manuscript publicized December 5, 2023.

[†]The authors are with Kyoto University, Kyoto-shi, 606–8501 Japan.
a) E-mail: dong.yihan.77v@st.kyoto-u.ac.jp

DOI: 10.1587/transinf.2023IHP0011

Copyright © 2024 The Institute of Electronics, Information and Communication Engineers

facilitation agent with the language learning model.

The structure of this paper is organized as follows: The second section presents a comprehensive design of the multi-phase facilitation agent, which is equipped to facilitate discussions across various stages. Following that, the third section details the combination with the Language Learning Model (LLM) and illustrates the additional functionalities enabled by the use of LLM. Subsequently, the fourth section provides specific details on the implementation, exhibiting screenshots to depict the operation of the multi-phase facilitation agent. The fifth section then provides an evaluation of it. Moreover, the sixth section engages in a discussion of the strengths and weaknesses associated with it. Finally, the seventh section offers a summary and concluding remarks for the paper.

2. Related Research

2.1 An Existing Facilitation Agent Based on IBIS

As mentioned previously, there is an IBIS-based agent, and it has been proved that it has better performance than human facilitators [7]. To explain how the IBIS-based agent works, the structure of IBIS will be introduced firstly: All the speeches in discussions can be classified into 4 different types, including issue, idea, pro and con. Since ideas aim to solve issues, pros and cons evaluate ideas, and new issues come from evaluations, there is a nested structure inside each discussion naturally [2]. The principle of the IBIS-based agent is simple: By classifying the types of existing posts, the agent generates facilitation messages according to the types of them to promote participants. For example, encouraging participants to give their ideas about an issue or to evaluate existing ideas, etc.

2.2 Definition of Inclusive Discussion Environment

As we pointed out as the first issue, in discussion environments without human facilitators, it is harder for shy participants to join in discussions while it is easier for some dominant participants to lead the discussion at their own pace. A survey figuring out the factors affecting the participation rates in class discussions shows that student traits such as confidence or comprehension, and discussion environment elements like interaction norms, are the two important factors [8]. Therefore, by quoting the definition of inclusion in work groups, which means that different diverse individuals could blend into the environment safely and harmoniously [9], [10], the discussion environments with the IBIS-based agent cannot say “inclusive” enough.

2.3 Large Language Models

The development of large language models (LLM) is also increasing dramatically in recent months. Since the impressive machine-learning network architecture, the Transformer has been raised for several years, multiple language models are also designed and trained to complete language tasks [11]. For example, BERT and GPT-3, which are the predecessors of ChatGPT were trained through this particular network to adjust to various natural language tasks [12], [13]. Because the usage of language models has illustrated excellent potential in various fields, we integrate LLM into the multi-phase facilitation agent with LLM.

2.4 Language Model Controlling Principles

There are two concepts called *Chain of Thought (CoT)* and *Action Plan Generation* are particularly attractive. The former concept was raised by Wei et al. and it has been proven that it can improve the performance of LLM in multiple kinds of tasks including arithmetic, commonsense, and symbolic reasoning tasks by applying LLM with this kind of principle [14], [15]. On the other hand, the latter focuses on feeding the responses generated from LLM back to LLM with evaluations of human beings to improve the performance of LLM in dealing with a particular kind of task [16], [17]. Both these two concepts influenced the design and implementation of a series of language model-controlling frameworks such as *LangChain* and *Guidance* from Microsoft. These concepts and tools are generic purposes while our multi-phase facilitation agent focuses on facilitation-specific purposes.

3. Design of Multiple Discussion Phases Control

First, the definitions of *facilitator* and *agent* are clarified. In the past, the facilitator and the agent often present multiple meanings, which include both generating the facilitation messages and the behaviours like creating a thread or replying to a post on platforms. However, both the facilitation messages and the behaviours at every phase vary at different phases. Therefore, in this paper, each phase has its own *facilitator*, which generates facilitation messages and decides behaviours. By contrast, the *agent* only represents the executor of these behaviours on platforms.

Secondly, the logical design and workflow of the multi-phase facilitation agent are explained. Before the detailed design explanation, we present a general description of the relationship among all modules in the multi-phase facilitation agent. As Fig. 1 illustrates, there are totally 7 main

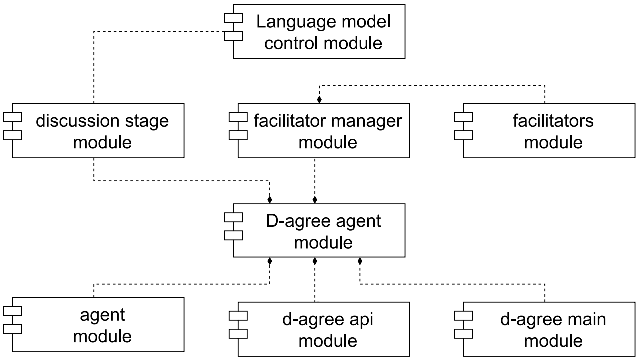


Fig. 1 The general description of the relationship among all modules

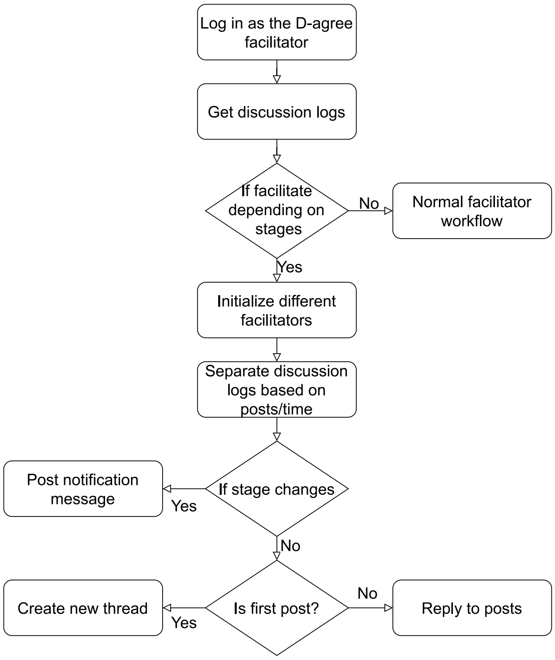

Log in as the D-agree facilitator
Get discussion logs
If facilitate depending on stages
No
Normal facilitator workflow
Yes
Initialize different facilitators
Separate discussion logs based on posts/time
If stage changes
Yes
Post notification message
No
Is first post?
Yes
Create new thread
No
Reply to posts

Fig. 2 The general workflow of the multi-phase facilitation agent

modules forming the whole system together. Particularly, the modules, *discussion phase, facilitators, facilitator manager, and language model control*, are the core parts to realize the functions. The basic idea of the general design is to separate modules from each other except the language model control module. Then, we make a particular module bounded with a specific platform, e.g. D-agree. This particular module represents the specific agent running on the D-agree platform, and it should include all the other modules to activate itself.

The whole workflow of the multi-phase facilitation agent is illustrated in Fig. 2. Depending on if the agent needs to facilitate due to different phases, the original workflow should be isolated from its workflow at first. Then, different combinations of facilitators belonging to their discussion phases are initialized, and the discussion logs are separated according to different phases. After that, if the phase last time is different from the current phase, then the agent posts a notification message. Finally, depending on whether it is the first time the facilitator of this phase posts a message, the behaviour of the facilitator changes respectively.

Finally, the class design of the multi-phase facilitation agent is demonstrated. Since there are multiple modes of discussion phases, it is significant and necessary to make the design flexible enough to adjust to extensions in the future. Thus, the basic idea is to build a *FacilitatorFactory* class and make all the detailed facilitator classes inherit from this class following the factory method design pattern [18]. The relationship among the classes mentioned above is illustrated in Fig. 3.

Furthermore, to make these facilitators react to the change in each discussion phase, we designed the following classes. The detailed facilitator classes are under the control of the class *FacilitatorManager*. The class *Discussionstage* manages a series of methods to deal with all the processes related to discussion phases like judging the current phase, separating participants' posts according to phases, and achieving participants' posts in the last phase. The related class relationship and the main methods are illustrated in Fig. 4.

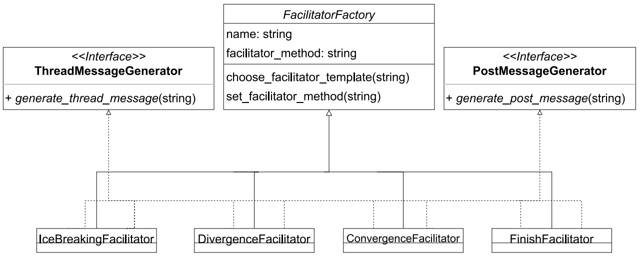


Fig. 3 The class diagram explaining how to generate various facilitators

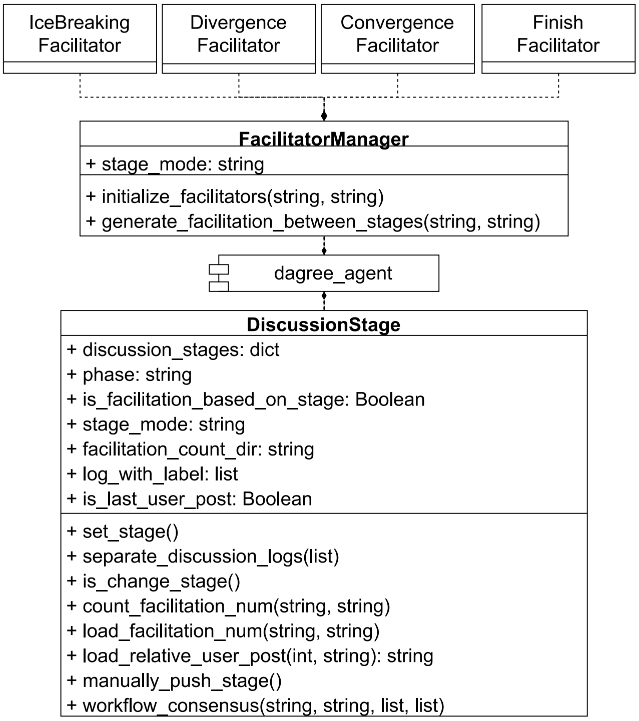

IceBreaking Facilitator
Divergence Facilitator
Convergence Facilitator
Finish Facilitator
FacilitatorManager
+ stage_mode: string
+ initialize_facilitators(string, string)
+ generate_facilitation_between_stages(string, string)
dagree_agent
DiscussionStage
+ discussion_stages: dict
+ phase: string
+ is_facilitation_based_on_stage: Boolean
+ stage_mode: string
+ facilitation_count_dir: string
+ log_with_label: list
+ is_last_user_post: Boolean
+ set_stage()
+ separate_discussion_logs(list)
+ is_change_stage()
+ count_facilitation_num(string, string)
+ load_facilitation_num(string, string)
+ load_relative_user_post(int, string): string
+ manually_push_stage()
+ workflow_consensus(string, string, list, list)

Fig. 4 The class diagram explaining the classes controlling facilitators in different phases and the classes managing discussion phases

4. Design of Combination with LLM

4.1 Design of Behaviours of the Multi-Phase Facilitation Agent

The design of how to connect the multi-phase facilitation agent with LLM to adjust to different language processing tasks is described in this section. To begin with, the detailed workflow of the agent in different phases is illustrated in Fig. 5.

As it illustrates, the multi-phase facilitation agent has two behaviours called *response* and *guidance* respectively. Besides, in each phase, it has a unique aim to promote partic-

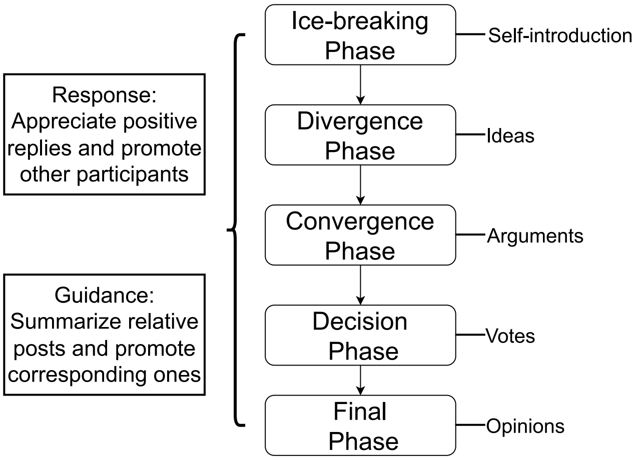

Response: Appreciate positive replies and promote other participants
Guidance: Summarize relative posts and promote corresponding ones
Ice-breaking Phase
Self-introduction
Divergence Phase
Ideas
Convergence Phase
Arguments
Decision Phase
Votes
Final Phase
Opinions

Fig. 5 The detailed flowchart of automated facilitation agent in different discussion phases

ipants to post different content. At the beginning of phases, it guides participants on what kinds of content are welcomed in this particular phase. Then, it responds to participants according to detailed content and encourages other participants to post similar content.

For example, during the ice-breaking phase, the multi-phase facilitation agent encourages participants to do self-introductions to get familiar with each other. Similarly, during the divergence phase, it encourages participants to give ideas related to the discussion topic; During the convergence phase, it encourages participants to raise arguments towards the ideas raised in the last phase alternatively; During the decision phase, it encourages participants to vote for their favourite ideas to reach a consensus; and it encourages participants to give their opinions finally.

Besides, to guide the participants better, in some phases like *convergence*, *decision* and *final*, the agent also summarizes the participants' posts in the previous phase. For example, it summarizes ideas which are raised by participants previously to remind participants and control the scale of the discussion in the convergence and decision phase. Alternatively, it summarizes the results of voting after the decision phase and illustrates them to participants in the final phase.

4.2 Design of Controlling LLM with Proper Prompts

To explain how to control the interactions with LLM, an explanation of the settings about parameters and prompts is necessary: The model used to generate facilitation messages is "gpt-3.5-turbo-16k", and all the other parameters are set as default.

To improve the quality of the responses received from LLM, the interaction with LLM is particularly designed, as Fig. 6 illustrates. The generation of facilitation messages is under the influence of previous posts.

The prompts to instruct the LLM on how to generate proper facilitation messages at the convergence phase are given as an example since all the prompts followed the same pattern as this example. At the beginning of the convergence phase, the LLM is given a task to summarize the previous posts at the divergence phase and the prompts are as follows: "Please summarize and classify the ideas from the previous participants' posts {previous ideas}, and print them out as Python string". After that, the summarized posts are given to the LLM as a part of the second task to select one idea from them and generate arguments about it. The prompts at this step are: "Please select one idea from previous participants' posts {summarized ideas}, and raise 1 advantage and 1 disadvantage of it considering the discussion topic of {discussion topic} within 20 words". Finally, the arguments generated by the LLM are provided again to encourage participants to raise more arguments related to the ideas raised in the last phase. The prompts of this task are listed as follows: "To inspire participants to raise advantages and disadvantages as the example related to the discussion topic {discussion topic}, with demonstrating the discussion stage right now is convergence, please generate a message within 50 words. The example of argument is as follows: {generated arguments}".

When the agent replies to a participant, to encourage other participants to raise arguments different from previous ones, the limitation of no-repeat is also added to the prompts as follows: "Please raise an argument different from the previous participants' posts {previous arguments} within 10 words".

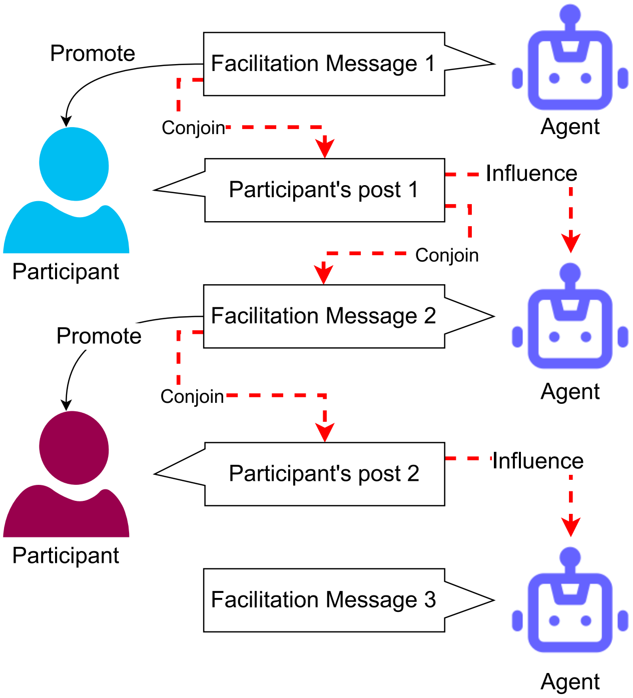


Fig. 6 The design of interactions with LLM in the multi-phase facilitation agent

5. Evaluation

5.1 Outline of Evaluations

This section evaluates the quality of the facilitation messages generated by the multi-phase facilitation agent, compared to the previous one which is based on IBIS [7]. The evaluation includes a questionnaire to investigate the imagination of different facilitation agents, and the calculation of two indexes to measure the diversity and the naturalness of facilitation agents. To make the comparison and evaluation clearer, the examples of facilitation messages generated by the IBIS-based agent and multi-phase facilitation agent in Japanese are illustrated in the following figures respectively. Particularly, Fig. 8 illustrates the IBIS-based agent responding to participants passively, but the guidance of the multi-phase facilitation agent in different phases is demonstrated in Figs. 9 and 10.

5.2 Comparison between IBIS-Based and Multi-Phase Facilitation Agents

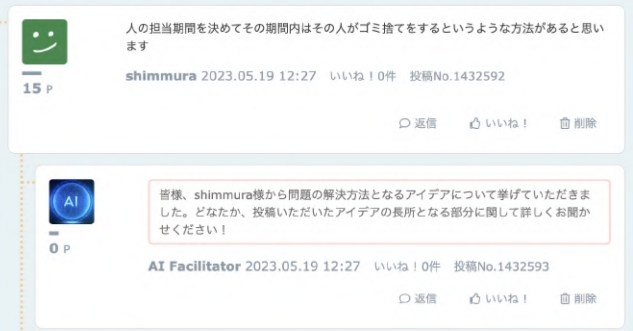
After the explanation of how the IBIS-based agent facilitates and the introduction of the design of the multi-phase facilitation agent, there are two advantages in design: The designs of facilitating according to different phases and combining with LLM. The comparison of facilitation messages will be shown as follows.

First, we show a facilitation message generated by the IBIS-based agent as an example. As Fig. 7 illustrates, the IBIS-based agent can only respond to existing posts. Meanwhile, it could not respond to participants' posts according to the contents, which is another drawback.

By contrast, as Fig. 8 and Fig. 9 illustrate, the multi-phase facilitation agent can guide the participants to join in the discussions with different contents at different phases. Furthermore, it can even create more corresponding and relative responses than the IBIS-based agent.

5.3 Evaluation Through a Case Study

As the evaluation of the automated facilitation agent that promotes human discussions, feedback from human beings is inevitable. Hence, the case study was held through two questionnaires including examples of the facilitation messages generated by the IBIS-based agent and the multi-phase facilitation agent respectively. In the questionnaire, five questions related to the performance of different automated facilitation



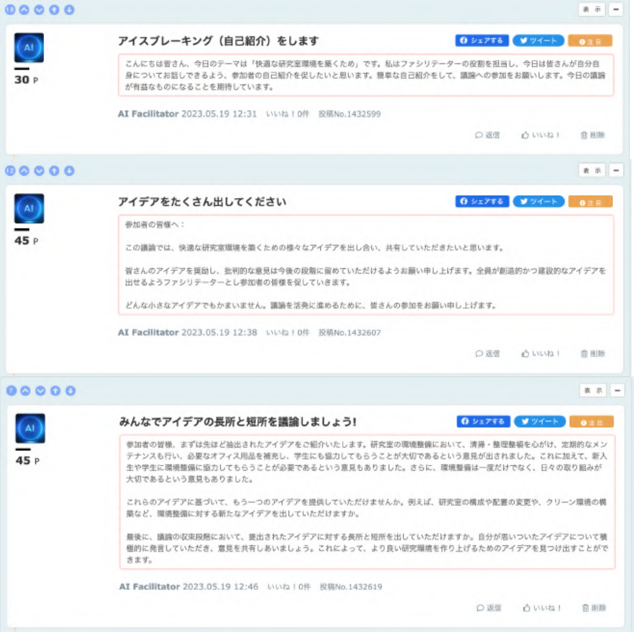
人の担当期間を決めてその期間内はその人がゴミ捨てをするというような方法があると思います

shimmura 2023.05.19 12:27 いいね！0件 投稿No.1432592

皆様、shimmura様から問題の解決方法となるアイデアについて挙げていただきました。どなたか、投稿いただいたアイデアの長所となる部分に関して詳しくお聞かせください！

AI Facilitator 2023.05.19 12:27 いいね！0件 投稿No.1432593

Fig. 7 An example of facilitation messages generated by IBIS-based agent



アイスブレーキング（自己紹介）をします

こんにちは皆さん、今日のテーマは「快適な研究室環境を築くため」です。私はファシリテーターの役割を担当し、今日は皆さんが自分自身についてお話しできるよう、参加者の自己紹介を促したいと思います。簡単な自己紹介をして、議論への参加をお願いします。今日の議論が有益なものになることを期待しています。

AI Facilitator 2023.05.19 12:31 いいね！0件 投稿No.1432599

アイデアをたくさん出してください

参加者の皆様へ：

この議論では、快適な研究室環境を築くための様々なアイデアを出し合い、共有していただきたいと思います。

皆さんのアイデアを奨励し、批判的な意見は今後の段階に留めていただけるようお願い申し上げます。全員が創造的かつ建設的なアイデアを出せるようファシリテーターとし参加者の皆様を促していきます。

どんな小さなアイデアでもかまいません。議論を活発に進めるために、皆さんの参加をお願い申し上げます。

AI Facilitator 2023.05.19 12:38 いいね！0件 投稿No.1432607

みんなでアイデアの長所と短所を議論しましょう!

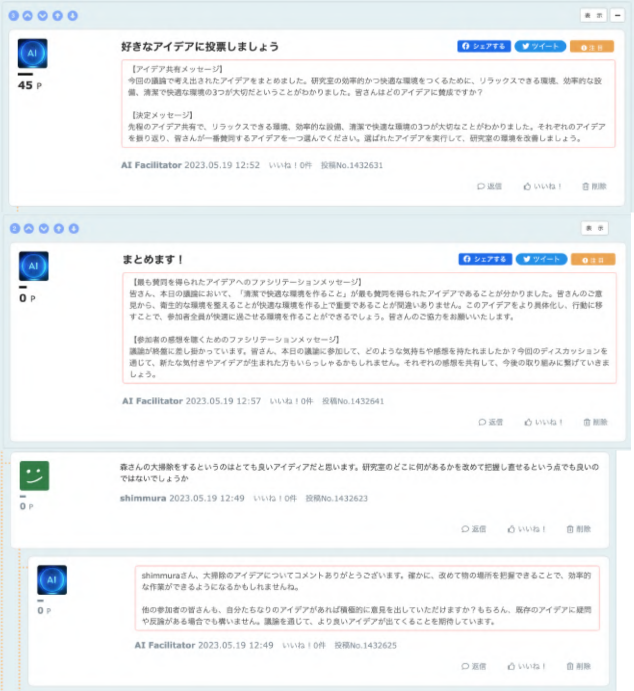
参加者の皆様、まずは先ほど抽出されたアイデアをご紹介いたします。研究室の環境整備において、清掃・整理整頓を心がけ、定期的なメンテナンスも行い、必要なオフィス用品を補充し、学生にも協力してもらうことが大切であるという意見が出されました。これに加えて、新入生や学生に環境整備に協力してもらうことが必要であるという意見もありました。さらに、環境整備は一度だけでなく、日々の取り組みが大切であるという意見もありました。

これらのアイデアに基づいて、もう一つのアイデアを提供していただけませんか。例えば、研究室の構成や配置の変更や、クリーン環境の構築など、環境整備に対する新たなアイデアを出していただけますか。

最後に、議論の収束段階において、提出されたアイデアに対する長所と短所を出していただけますか。自分が思いついたアイデアについて積極的に発言していただき、意見を共有しあいましょう。これによって、より良い研究環境を作り上げるためのアイデアを見つけ出すことができます。

AI Facilitator 2023.05.19 12:46 いいね！0件 投稿No.1432619

Fig. 8 Examples of facilitation messages at the ice-breaking, divergence and convergence stage



好きなアイデアに投票しましょう

【アイデア共有メッセージ】
今回の議論で考え出されたアイデアをまとめました。研究室の効率的かつ快適な環境をつくるために、リラックスできる環境、効率的な設備、清潔で快適な環境の3つが大切だということがわかりました。皆さんはどのアイデアに賛成ですか？

【決定メッセージ】
先程のアイデア共有で、リラックスできる環境、効率的な設備、清潔で快適な環境の3つが大切なことがわかりました。それぞれのアイデアを振り返り、皆さんが一番賛同するアイデアを一つ選んでください。選ばれたアイデアを実行して、研究室の環境を改善しましょう。

AI Facilitator 2023.05.19 12:52 いいね！0件 投稿No.1432631

まとめます！

【最も賛同を得られたアイデアへのファシリテーションメッセージ】
皆さん、本日の議論において、「清潔で快適な環境を作ること」が最も賛同を得られたアイデアであることが分かりました。皆さんのご意見から、衛生的な環境を整えることが快適な環境を作る上で重要であることが間違いありません。このアイデアをより具体化し、行動に移すことで、参加者全員が快適に過ごせる環境を作ることができるでしょう。皆さんのご協力をお願いいたします。

【参加者の感想を聴くためのファシリテーションメッセージ】
議論が終盤に差し掛かっています。皆さん、本日の議論に参加して、どのような気持ちや感想を持たれましたか？今回のディスカッションを通じて、新たな気付きやアイデアが生まれた方もいらっしゃるかもしれません。それぞれの感想を共有して、今後の取り組みに繋げていきましょう。

AI Facilitator 2023.05.19 12:57 いいね！0件 投稿No.1432641

森さんの大掃除をするというのはとても良いアイディアだと思います。研究室のどこに何があるかを改めて把握し直せるという点でも良いのではないでしょうか

shimmura 2023.05.19 12:49 いいね！0件 投稿No.1432623

shimmuraさん、大掃除のアイデアについてコメントありがとうございます。確かに、改めて物の場所を把握できることで、効率的な作業ができるようになるかもしれませんね。

他の参加者の皆さんも、自分たちなりのアイデアがあれば積極的に意見を出していただけますか？もちろん、既存のアイデアに疑問や反論がある場合でも構いません。議論を通じて、より良いアイデアが出てくることを期待しています。

AI Facilitator 2023.05.19 12:49 いいね！0件 投稿No.1432625

Fig. 9 An example of facilitation messages at the decision and finish stage

Table 1 The evaluation results through distinct-1 score and PLL score

	average token number per sentence	distinct-1 score	PLL score
original IBIS-based agent	21.72	0.1016	−44.14
IBIS-based agent with gpt	24.48	0.1206	−43.78
multi-phase facilitation agent	21.11	0.2134	−36.08

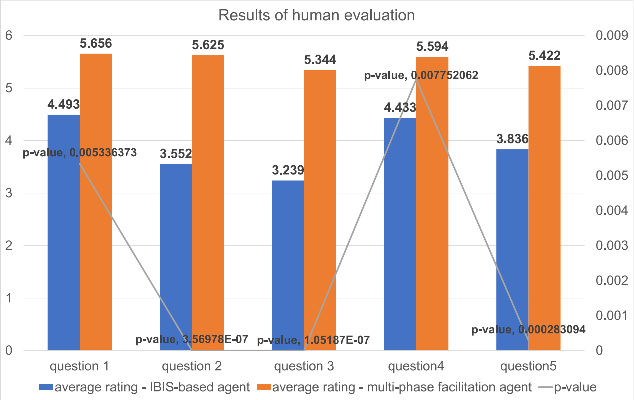


Fig. 10 The results of the evaluation through the case study

agents are included: *1) How natural are the facilitation messages? 2) How diverse are the facilitation messages? 3) Do you feel that the agents were speaking on their own initiative? 4) Are the facilitation messages easy to understand? and 5) Do you want to follow the agent's instructions?* For each question, the respondents expressed their feedback with a rating from 1 to 10 according to their degree of support.

The case study was conducted on a group of first-year undergraduate students in a university, with the participation of several students belonging to another university from fourth-year undergraduates to second-year postgraduates. Since the sample sizes of respondents attending the evaluation of two automated facilitation agents are not exactly the same and the variances are unequal, Welch's t-test is used to validate if there are significant differences between the results of the two questionnaires [19].

As Fig. 10 shows, the average ratings of the multi-phase facilitation agent received from the respondents are far higher than those of the IBIS-based agent. Besides, since each *p*-value is small enough, the conclusion that the evaluations towards these two automated facilitation agents have significant differences can be reached.

5.4 Evaluation Through Distinct-N and PLL

To support the results of the case study, two more evaluation standards are selected to judge which automated facilitation agent is better: One is called *distinct-N* to evaluate the diversity of the content generated by LLMs [20]; Another is called the pseudo-log-likelihood scores (PLLs) to evaluate the linguistic correctness of the content generated by LLMs [21].

Distinct-N is the ratio of the number of unique tokens to that of the total tokens in the N-gram of text generated by LLMs. The formula to calculate distinct-N is as follows:

$$distinct - N = \frac{distinct\ N{-}gram\ numbers}{total\ N{-}gram\ numbers} \tag{1}$$

According to the formula above, the contents generated by LLMs with higher distinct-N have more diversity and it can be concluded as a standard to judge which automated facilitation agent can generate more diverse facilitation messages. Since the evaluation using this stand to measure the diversity of facilitation messages only counts the unique single token in sentences, the rating is also called and illustrated as distinct-1.

On the other hand, PLLs are computed by masking tokens one by one. Literally, PLLs are the sum of the log-likelihoods of the conditional probabilities of predicting each word hidden with a mask [21].

Compared to other measuring standards like ROUGE [22] or BLEU [23], one of the most vital advantages of using PLLs is that, PLLs do not need reference text to judge whether the contents generated by LLMs are correct or not. PLLs evaluate the correctness of contents with a minus score and the contents with a higher score are more correct. Since it is relatively hard to make standards for facilitation messages, PLLs are selected to evaluate facilitation messages generated by automated facilitation agents due to their unique advantage.

Since the IBIS-based agent is designed without the use of LLM, the facilitation messages are based on template files according to different scenarios. Thus, we also used gpt-3.5 to improve the facilitation messages with the following prompts: "Here is the list of facilitation messages. Not all the facilitation messages are fluent enough. Please modify and improve those facilitation messages are not good enough and reply them to me". The comparison is among the original IBIS-based agent, the IBIS-agent agent with improved facilitation messages by gpt and the multi-phase facilitation agent. The results of the evaluation are demonstrated in Table 1.

According to the evaluation results through distinct-1 and PLL scores, it is evident that the multi-phase facilitation agent has about twice the diversity in the facilitation messages and far more natural contents in the sentences with almost the same length of sentences compared to the IBIS-based agent.

6. Discussion

6.1 Comparison with IBIS-Based Agent

Based on the evaluation results no matter whether according to human feedback or other evaluation standards, it can be concluded that the multi-phase automated facilitation agent has a great advantage in promoting participants with different characteristics to join in discussions. Particularly, according to the results of the case study, the facilitation messages generated by the multi-phase facilitation agent are believed to be easier to understand, with the features of speaking on their initiative. Meanwhile, respondents also expressed that they are more likely to follow the instructions of the multi-phase facilitation agent in discussions.

On the other hand, based on the evaluation results using distinct-1 and PLL scores, even though gpt improves the performance of the original IBIS-based agent, the multi-phase facilitation agent still has an overwhelming advantage in both diversity and naturalness. The reason is that the original IBIS-based agent generates facilitation messages through template files. By contrast, the facilitation messages of the multi-phase agent are generated based on both phases and users' posts. Even though the template files are upgraded using LLM, the facilitation messages of the IBIS-based agent are still relatively unnatural and lack of diversity.

Besides, there is also improvement in the structure of the framework. As mentioned above, flexibility is fully considered during the design and implementation. Literately, part of the design and the implementation can be seen as the refactoring of the IBIS-based agent.

6.2 Comparison with Another LLM Application Framework

Except for the comparison with the IBIS-based agent, the comparison with the Python library called *LangChain* in the structure of projects also should be raised in this section since LangChain was designed and implemented based on the principles of *Chain of Thought*, *Action Plan Generation* and other concepts.

LangChain is a sort of framework connecting with LLM to build agents that can call multiple language models to make them learn from real-time communication to generate responses with relatively higher quality. As mentioned in the official document of LangChain, the evaluation and comparison towards different kinds of agents since there are no proper data sets and metrics for evaluation. Therefore, the comparison with LangChain is based on the difference in the structure.

Even though LangChain has sufficient functions and APIs to create customized LLM agents with high-level performance, there is a significant difference between LangChain and the framework raised in this paper: Since the aim of our framework is to create facilitation agent promoting discussions, and the changes of discussion phases were fully considered previously, the implementation of dividing discussions into multiple phases and behaving according to different phases based on our framework is far simpler than that with LangChain. Besides, by relying on this framework, multiple facilitation agents for discussions with different aims also can be easily created. By contrast, it is possible to use LangChain to achieve similar targets though, it would cost more time due to the lack of standard frameworks.

6.3 Discussion Summary

According to the results of the evaluations mentioned above, there are several conclusions that can be reached:

Compared to the IBIS-based agent, the multi-phase facilitation agent implemented based on the framework has overwhelming advantages in the diversity and naturalness of facilitation messages, both admitted by human feedback and automatic test standards. Even though the multi-phase facilitation agent has the disadvantage that it cannot identify the type of participants' posts, it still has better performance than the IBIS-based agent.

On the other hand, compared to LangChain, the feature of promoting discussions according to different phases is the most valuable advantage of the proposed facilitation framework, while LangChain can work in general-purpose applications.

7. Conclusions

This paper demonstrates the design and implementation of a phase-based facilitation framework based on LLM to realize inclusive discussion environments and the efficient use of an LLM to facilitate realistic discussions. Furthermore, to validate that the multi-phase facilitation agent implemented based on the framework has advantages compared to the IBIS-based agent, this paper raises the results of evaluation through human feedback and automatic evaluation standards. Both results illustrate that the facilitation messages generated by the multi-phase facilitation agent are better than those generated by the IBIS-based agent in diversity and naturalness. Finally, this paper discusses the advantages and disadvantages of this framework compared with the IBIS-based agent and LangChain.

However, it does not mean that the multi-phase facilitation agent has no disadvantages. One of the most vital disadvantages is that without the combination with other fine-tuned language models, it cannot identify the type of previous posts by a particular label. Hence, the participants' posts can only be selected by which phase they were raised, but not a clear classification result. Also, it can only switch to another discussion phase depending on the time or the number of participants' posts, but not both. To improve these points is in our future work. To further improve the quality of discussions, other kinds of agents will be designed and implemented based on this framework in the future.

Acknowledgements

This work was supported by JST CREST Grant Number JPMJCR20D1, Japan and JST, the establishment of university fellowships towards the creation of science technology innovation, Grant Number JPMJFS2123.

References

[1] T. Ito, S. Suzuki, N. Yamaguchi, T. Nishida, K. Hiraishi, and K. Yoshino, "D-agree: crowd discussion support system based on automated facilitation agent," Proceedings of the AAAI conference on artificial intelligence, vol.34, no.9, pp.13614–13615, 2020.

[2] D.E. Noble and H.W.J. Rittel, "Issue-based information systems for design," Univ. Calif. Berkeley Work. Pap, vol.492, 1989.

[3] R. Hadfi and T. Ito, "Exploring Interaction Hierarchies in Collaborative Editing using Integrated Information," Collective Intelligence Conference, ACM, 2021.

[4] R. Hadfi, J. Haqbeen, S. Sahab, and T. Ito, "Argumentative conversational agents for online discussions," Journal of Systems Science and Systems Engineering, vol.30, pp.450–464, 2021.

[5] J. Haqbeen, T. Ito, R. Hadfi, T. Nishida, Z. Sahab, S. Sahab, S. Roghmal, and R. Amiryar, "Promoting discussion with ai-based facilitation: Urban dialogue with kabul city," Proceedings of the 8th ACM Collective Intelligence, ACM Collective Intelligence Conference Series, Boston (Virtual Conference), South Padre Island, TX, USA, 2020.

[6] J. Haqbeen, T. Ito, S. Sahab, R. Hadfi, S. Okuhara, N. Saba, M. Hofaini, and U. Baregzai, "A contribution to covid-19 prevention through crowd collaboration using conversational ai & social platforms," arXiv preprint arXiv:2106.11023, 2021.

[7] T. Ito, R. Hadfi, and S. Suzuki, "An agent that facilitates crowd discussion: A crowd discussion support system based on an automated facilitation agent," Group Decision and Negotiation, vol.31, no.3, pp.621–647, 2022.

[8] J.E. Aitken and M.R. Neer, "The relationship of classroom communication apprehension and motivation to college student question-asking.," 1992.

[9] L.M. Shore, A.E. Randel, B.G. Chung, M.A. Dean, K. Holcombe Ehrhart, and G. Singh, "Inclusion and diversity in work groups: A review and model for future research," Journal of management, vol.37, no.4, pp.1262–1289, 2011.

[10] V. Grubbs, "Diversity, equity, and inclusion that matter," New England Journal of Medicine, vol.383, no.4, p.e25, 2020.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," Advances in neural information processing systems, vol.30, 2017.

[12] J. Devlin, M.W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," arXiv preprint arXiv:1810.04805, 2018.

[13] T. Brown, B. Mann, N. Ryder, M. Subbiah, J.D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., "Language models are few-shot learners," Advances in neural information processing systems, vol.33, pp.1877–1901, 2020.

[14] J. Wei, X. Wang, D. Schuurmans, M. Bosma, E. Chi, Q. Le, and D. Zhou, "Chain of thought prompting elicits reasoning in large language models," arXiv preprint arXiv:2201.11903, 2022.

[15] M. Nye, A.J. Andreassen, G. Gur-Ari, H. Michalewski, J. Austin, D. Bieber, D. Dohan, A. Lewkowycz, M. Bosma, D. Luan, et al., "Show your work: Scratchpads for intermediate computation with language models," arXiv preprint arXiv:2112.00114, 2021.

[16] R. Nakano, J. Hilton, S. Balaji, J. Wu, L. Ouyang, C. Kim, C. Hesse, S. Jain, V. Kosaraju, W. Saunders, et al., "Webgpt: Browser-assisted question-answering with human feedback," arXiv preprint arXiv:2112.09332, 2021.

[17] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, et al., "Do as i can, not as i say: Grounding language in robotic affordances," arXiv preprint arXiv:2204.01691, 2022.

[18] M. Summerfield, Python in practice: create better programs using concurrency, libraries, and patterns, Addison-Wesley, 2013.

[19] B.L. Welch, "The generalization of 'student's' problem when several different population varlances are involved," Biometrika, vol.34, no.1-2, pp.28–35, 1947.

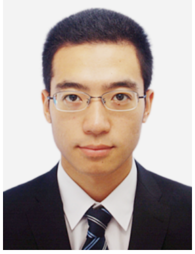
[20] J. Li, M. Galley, C. Brockett, J. Gao, and B. Dolan, "A diversity-promoting objective function for neural conversation models," arXiv preprint arXiv:1510.03055, 2015.

[21] J. Salazar, D. Liang, T.Q. Nguyen, and K. Kirchhoff, "Masked language model scoring," arXiv preprint arXiv:1910.14659, 2019.

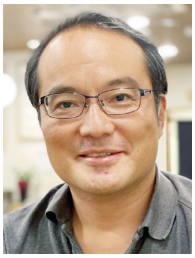
[22] C.Y. Lin, "Rouge: A package for automatic evaluation of summaries," Text summarization branches out, pp.74–81, 2004.

[23] C.-Y. Lin and E. Hovy, "Automatic evaluation of summaries using n-gram co-occurrence statistics," Proceedings of the 2003 human language technology conference of the North American chapter of the association for computational linguistics, pp.150–157, 2003.

Yihan Dong is a PhD student at the Graduate School of Informatics at Kyoto University, Japan. He received his Master's degree in Engineering(Software) at the University of Melbourne, Australia in 2021.



Shiyao Ding is an assistant professor in the Graduate School of Informatics from Kyoto University, Japan. He received his Master's degree in engineering from Osaka University, Japan in September 2019 and his PhD degree from Kyoto University, Japan in September 2022. His current research interests include reinforcement learning, graph neural networks, multiagent systems, argumentation mining and services computing.



Takayuki Ito is a professor and head of the Department of Social Informatics at Kyoto University. He received his Doctor Degree in Engineering from the Nagoya Institute of Technology in 2000. He was a JSPS research fellow, an associate professor of JAIST, and a visiting scholar at USC/ISI, Harvard University, and MIT twice. He was a board member of IFAAMAS, the PCchair of AAMAS2013, PRIMA2009, GeneralChair of PRIMA2014, IEEE ICA2016, is the Local Arrangements Chair of IJCAI2020, and was an SPC/PC member in many top-level conferences (IJCAI, AAMAS, ECAI, AAAI, etc). He received the JSAI Contribution Award, the JSAI Achievement Award, the JSPS Prize, the Fundamental Research Award of JSSST, the Prize for Science and Technology of the Commendation for Science and Technology by the Minister of Education, Culture, Sports, Science, and Technology (MEXT), the Young Scientists' Prize of the Commendation for Science and Technology by the MEXT, the Nagao Special Research Award of IPSJ, the Best Paper Award of AAMAS2006, the 2005 Best Paper Award of JSSST, and the Super Creator Award of 2004 IPA Exploratory Software Creation Project. He was a JST PREST Researcher and a principal investigator of the Japan Cabinet Funding Program for Next Generation World-Leading Researchers. He is currently the principal investigator of his 2nd JST CREST project.