DiagGPT: An LLM-based Dialogue System with Automatic Topic Management for Goal-Oriented Dialogue

Anonymous ACL submission

Abstract

Large Language Models (LLMs), such as ChatGPT, are becoming increasingly sophisticated, demonstrating capabilities that closely resemble those of humans. These AI models are playing an essential role in assisting humans with a wide array of tasks in daily life. A significant application of AI is its use as a chat agent, responding to human inquiries across various domains. Current LLMs have shown proficiency in answering general questions. However, basic question-answering dialogue often falls short in complex diagnostic scenarios, such as legal or medical consultations. These scenarios typically necessitate Goal-Oriented Dialogue (GOD), wherein an AI chat agent needs to proactively pose questions and guide users towards specific goals or task completion. Previous fine-tuning models have underperformed in GOD, and current LLMs do not inherently possess this capability. In this paper, we introduce DiagGPT (Dialogue in Diagnosis GPT), an innovative method that extends LLMs to GOD scenarios. Our experiments reveal that DiagGPT exhibits outstanding performance in conducting GOD with users, demonstrating its potential for practical applications in various fields.

1 Introduction

Large language models (LLMs), such as ChatGPT, have demonstrated remarkable performance on various natural language processing (NLP) tasks (Brown et al., 2020; Chowdhery et al., 2022; Wei et al., 2022a; OpenAI, 2023). Leveraging large-scale pre-training on massive text corpora and reinforcement learning from human feedback (RLHF), LLMs not only possess a wide range of knowledge but also exhibit superior capabilities in language understanding, generation, interaction, and reasoning. In many cases, OpenAI GPT-4 even outperforms human performance (OpenAI, 2023). With the use of prompt engineering techniques (e.g., chain-of-thought prompting (Brown et al., 2020; Wei et al., 2022b), in-context learning (Brown et al., 2020; Xie et al., 2022; Min et al., 2022), etc.), we can unlock the unlimited potential of LLMs to complete complex tasks in our daily life. LLMs have attracted enormous attention from both academia and industry, inspiring more people to build fantastic applications based on them.

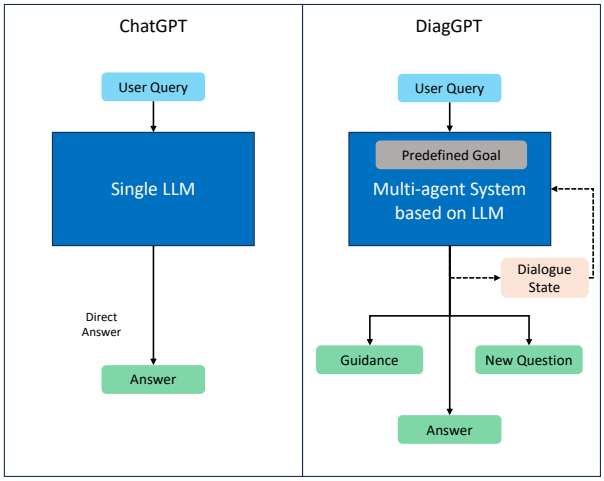

Figure 1: The main difference between ChatGPT and DiagGPT. While ChatGPT directly answers user questions, DiagGPT not only provides answers of the same quality but also has the ability to proactively ask questions, guide users, and maintain dialogue state internally.

One popular application of LLMs is in chatbots, which build conversational systems around these models. ChatGPT[1] is a successful example of such an application, where the AI model has the ability to analyze context and respond to user queries based on knowledge derived from extensive training data. By supplementing its background knowledge and providing context and appropriate prompts, ChatGPT has been able to form robust question-answering models for specialized fields. It can understand users' questions and provide precise answers effectively.

[1] https://openai.com/blog/chatgpt

However, dialogue scenarios in our daily life can be more complex. For instance, in specialized professional consultation scenarios like legal or medical diagnosis, the AI model needs to consider the user's unique situation or information. In the process of obtaining user information, the interactive experience provided by the AI model is also crucial. The system need to proactively ask questions. Therefore, we need a consultation process from the AI model that better simulates real medical experts and legal professionals. The AI model should conduct question-answering, topic management, and guiding users towards specific goals or task completion. This type of dialogue is known as Goal-Oriented Dialogue (GOD). There are usually some predefined goals in a conversation. GOD helps users achieve their specific goals, focusing on understanding users, tracking states, and generating next actions (Balaraman et al., 2021). It is substantially different from light-conversational scenarios. Despite much research in this area, it remains challenging due to issues such as a lack of training data, inefficiency, and drawbacks of fine-tuning small models, including an inability to fully understand user meaning and poor generative performance. Models from the existing research on this topic is not robust and universal. For example, fine-tuning models require a lot of data to train and are difficult to transfer to other scenarios. On the other side, although LLMs have a wide range of knowledge and the quality of their answer is far beyond that of fine-tuning models, traditional LLMs no longer meet needs of GOD and cannot effectively manage complex dialogue logic. Because they maintain a simple memory base and can only handle linear interaction.

Recent advancements have focused on using LLMs as AI agents to construct multi-agent systems or to teach AI how to use tools to accomplish more complex tasks (Schick et al., 2023; Shen et al., 2023). These systems typically have a core AI agent that controls the entire task process. A prominent example is AutoGPT[2], which employs multiple GPT models to strategize the responsibilities of each agent in order to split complex tasks and then complete them. In such multi-agent systems, the key lies in the division of tasks and the interaction between agents. By organizing multiple LLMs and instruct them to collaborate, we have the opportunity to tackle many complex tasks which one LLM cannot do well.

Fine-tuning models fall short in terms of scenario transfer ability and training data requirements, while a single large language model is not proficient in dialogue state tracking and management. However, we can leverage the strong knowledge background of LLMs and employ a multi-agent framework to incorporate the dialogue state tracking and management ideas from fine-tuning models. Thus, we can construct a multi-agent dialogue system that fulfills the requirements of goal-oriented dialogue. Motivated by these considerations, we propose `DiagGPT` in this paper. `DiagGPT` stands for **Dia**log**u**e in **Diag**nosis model based on **GPT**-4. This is a multi-agent AI system, which has automatic topic management ability to enhance its utility in goal-oriented dialogue scenarios. In summary, our AI system `DiagGPT` possesses the following features:

- **Question Answering:** It is the distinctive feature of traditional LLM-based conversational systems. LLMs possess a wide range of knowledge and can provide high-quality answers to various questions. In our system, we retain this fundamental ability of LLMs.

- **Task Guidance:** The system is designed to guide users towards a specific goal and assist them in accomplishing the task throughout the dialogue progression. This is achieved by advancing a sequence of predefined topics throughout the dialogue.

- **Proactive Asking:** The system has the ability to proactively pose questions based on a predefined checklist, thereby collecting necessary information from users.

- **Topic Management:** The system is capable of automatically managing topics throughout the dialogue, tracking topic progression, and effectively engaging in discussions centered around the current topic. It performs well in managing various topic changes in complex dialogues.

- **Versatile:** Our system is directly based on LLMs. It can perform well in various scenarios without requiring any training data, a capability that previous fine-tuning models lack. This system can be easily applied to multiple cases by defining specific predefined

[2]https://github.com/Significant-Gravitas/Auto-GPT

goals and supplementing functions to support them.

- **High Extendibility:** In this paper, we only introduce the basic framework of this AI system aimed at achieving goal-oriented dialogue. We have designed the system with ample flexibility to incorporate additional functions to handle tasks in complex scenarios and to meet more needs of conversational systems. Besides, as the foundation model is upgraded, the performance of our system will also become better.

Given these features, DiagGPT can meet the aforementioned needs and better engage in professional consultation conversations with users in complex scenarios. Our main contribution is to make traditional LLM-based conversational systems smarter. We build on the strong knowledge of LLMs and give them more interactive capabilities. Therefore, DiagGPT can function like a more intelligent and more professional chatbot.[3]

2 Related Works

Goal-Oriented Dialogue systems assist users in achieving specific goals, focusing on understanding users, tracking states, and generating subsequent actions. Task-Oriented Dialogue (TOD) is a similar task in which the goal is to accomplish a specific objective. Recent work primarily focusing on fine-tuning small models. (Wen et al., 2017) introduce a neural network-based text-in, text-out end-to-end trainable goal-oriented dialogue system along with a new way of collecting dialogue data based on a novel pipe-lined Wizard-of-Oz framework. (Wu et al., 2019) propose a Transferable Dialogue State Generator (TRADE) that generates dialogue states from utterances using copy mechanism, facilitating transfer when predicting (domain, slot, value) triplets not encountered during training. (Feng et al., 2023) propose SG-USM, a novel schema-guided user satisfaction modeling framework. It explicitly models the degree to which the user's preferences regarding the task attributes are fulfilled by the system for predicting the user's satisfaction level. (Liu et al., 2023) propose a framework called MUST to optimize TOD systems via leveraging Multiple User Simulator. (Bang et al., 2023) propose an End-to-end TOD system with Task-Optimized Adapters which learn independently per task, adding only small number of parameters after fixed layers of pre-trained network. All these methods require a considerable amount of data for training and have not yet attained a performance level that is ideal for real-world applications.

Conversational Systems with LLMs have become popular as the robust capabilities of LLMs have been recognized. (Hudeček and Dušek, 2023) evaluated the conversational ability of LLMs and found that, in explicit belief state tracking, LLMs underperform compared to specialized task-specific models. This suggests that simple LLMs do not have the ability to achieve goal-oriented dialogue. (Liang et al., 2023) proposed an interactive conversation visualization system called C5, which includes Global View, Topic View, and Context-associated QA View to better retain contextual information and provide comprehensive responses. From another perspective, (Zhang et al., 2023) proposed the Ask an Expert framework in which the model is trained with access to an expert whom it can consult at each turn. This framework utilizes LLMs to improve fine-tuning small models in GOD. There is minimal work on improving the conversational ability of LLMs. To the best of our knowledge, we are the first to successfully use off-the-shelf LLMs in a multi-agent framework to build a goal-oriented dialogue system.

3 Methodology

3.1 DiagGPT Framework

DiagGPT is a multi-agent and collaborative system composed of several modules: *Chat Agent*, *Topic Manager*, *Topic Enricher*, and *Context Manager*. Each module is a LLM with specific prompts that guide their function and responsibility. Among these modules, the *Topic Manager* is particularly important as it tracks the dialogue state and automatically manages the dialogue topic.

As shown in Figure 2, the workflow of DiagGPT consists of four stages: 1) Thinking Topic Development: *Topic Manager* obtain the user query, then analyze and predict the topic development in current round of dialogue; 2) Maintaining Topic Stack: maintain the topic stack of the entire dialogue according to action commands from *Topic Manager*; 3) Enriching Topic: retrieve the current topic and

[3]Our code can be found in the supplementary material and will be made publicly available following the paper's acceptance.

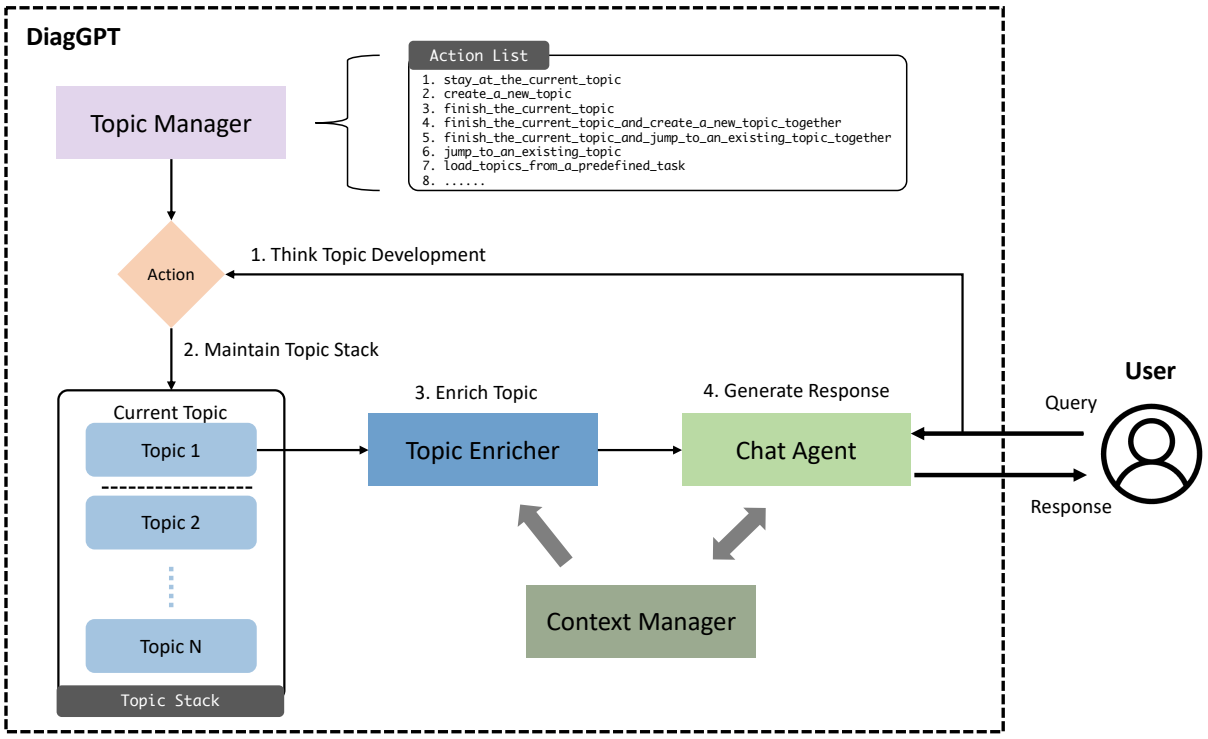

Figure 2: The framework of DiagGPT. The workflow of DiagGPT consists of four stages: Thinking Topic Development, Maintaining Topic Stack, Enriching Topic, Generating Response.

enrich it based on dialogue context; 4) Generating Response: based on specific guidance prompt and combined it with enriched topic and context to generate response for users.

Besides, we define a topic as the main subject of a round of dialogue, which determines the primary focus of communication. We also define a task as a specific goal that needs to be completed in a goal-oriented dialogue. After going through all the predefined topics in a dialogue, this specific task should be accomplished.

3.2 Thinking Topic Development

Topic Manager serves as the main module in DiagGPT and is responsible for determining the topic development based on the user's query. In each round of dialogue, the system needs to adjust the current dialogue topic before providing its response. Therefore, the user's query is first fed into *Topic Manager*.

The input to the *Topic Manager* includes the current user query, action list, the current status of the topic stack, and the chat history. It is logical for an AI agent to analyze and predict the topic development based on this information. Of particular importance is the action list stored in the *Topic Manager*. This action list contains various actions that serve as tools for the *Topic Manager* to execute. The *Topic Manager* has knowledge about the details of each action, how to plan and execute them. Each action corresponds to a program function that executes a specific command. In Python, we use decorator functions to implement this. Whenever *Topic Manager* receives a user query, it analyzes all the available information and decides which action to execute based on the prompts associated with each action.

With the strong understanding and reasoning abilities of LLMs, this AI agent can accurately comprehend the user's intentions and help to effectively engage in communication with the user.

3.3 Maintaining Topic Stack

After obtaining the output of the action from the *Topic Manager*, the system will execute the corresponding command to process and control the topic change, which involves maintaining the topic stack.

The topic stack is a data structure in this AI system that stores and tracks the dialogue state. We consider the progress of a dialogue to have multiple stages or states, and these states follow a first-in, first-out (FIFO) order, which can be effectively modeled using a stack. Although we refer to this

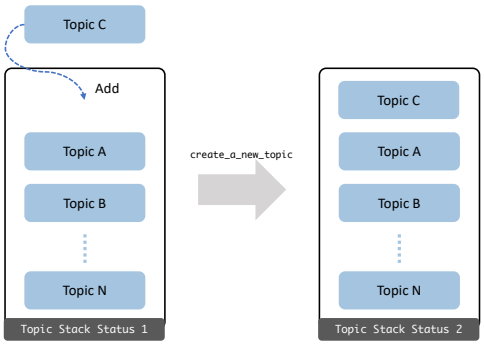

Figure 3: The action of creating a new topic.

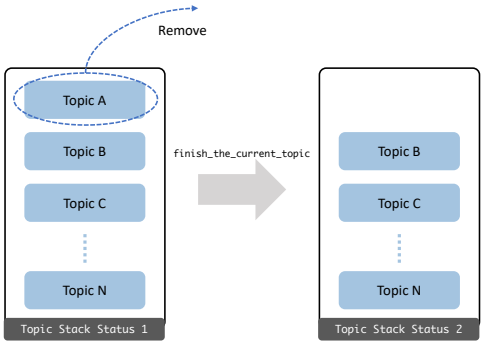

Figure 4: The action of finishing the current topic.

topic storage structure as a stack, this component does not strictly adhere to the FIFO rule. This is due to the presence of complex dialogue logic, and FIFO is merely the most common form of topic development. We will provide a detailed description of some operations of this structure in the following paragraphs.

In a diagnosis scenario, a consultant typically has a checklist stored in their mind. In many common cases, if users do not propose any new questions, the dialogue development will follow this checklist. After going through all the items in the checklist, the consultant can provide reports and comprehensive analysis to the users and complete the specific task. The action *load topics from a predefined task* is designed to facilitate this process. When the function decorated by this action prompt is executed, a list of topics from the checklist, will be loaded into the topic stack.

Furthermore, there are other actions commonly used to manipulate the topic stack. These actions include *create a new topic*, *finish the current topic*, and *stay at the current topic*. The *create a new topic* action, as shown in Figure 3, adds a new topic to the stack when the user wants to start a new topic. The *finish the current topic* action, shown in Figure 4, removes the top topic from the stack

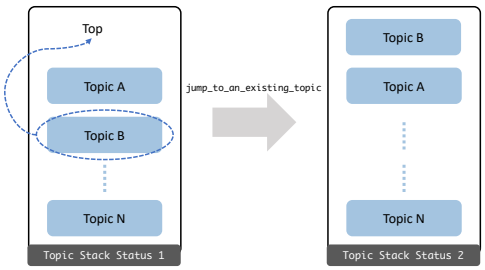

Figure 5: The action of jumping to an existing topic.

when the user no longer wishes to discuss it or the system considers this topic to be closed. The *stay at the current topic* action indicates that the system determines that it still requires information and needs continuous discussing the current topic, so the topic stack does not change at all. These three basic operations cover most topic change scenarios. Since we only allow one-step reasoning for LLMs, they must select and execute only one action. Other actions are complex changes based on these three basic operations.

The *jump to an existing topic* action, illustrated in Figure 5, allows the user to retrieve and prioritize a previous topic from the stack.

This action list can be expanded to accommodate more complex scenarios in goal-oriented dialogues. We have also implemented a mechanism to automatically remove redundant topics. After several rounds of dialogue, if a newly generated topic is not recalled, it will be removed. However, this removal does not affect any predefined topics from checklist.

3.4 Enriching Topic

We select the top item in the topic stack as the current topic. However, it cannot be directly used as a chat topic for the *Chat Agent* to interact with the user. The *Topic Enricher* is designed to bridge this gap and assist in better organizing the language for use. We initially categorize the topic into *Ask user* and *Answer user*. Typically, newly generated topics fall under *Answer user*, while predefined topics are categorized as *Ask user*. This distinction helps the system determine whether to answer a user's question or ask a question in the current round of dialogue. The *Topic Enricher* takes the output of the *Context Manager* and the current topic to enrich it into a topic that contains ample information and is contextually appropriate. This enriched topic is then provided to the *Chat Agent*.

3.5 Generating Response

With the final topic, the *Chat Agent* recognizes it as the primary topic in this round of dialogue. Thus, with context from the *Context Manager*, it can finally generate responses for users. In addition, as shown in Figure 9, some retrieved background knowledge, instructions, and encouragements will also be added into prompts here to further improve the response quality.

3.6 Extendibility

We have only extracted the most important modules in `DiagGPT` to form a basic framework of a system. These four modules can already implement the basic functions of goal-oriented dialogue. In a multi-agent AI system, there is a large extendibility. For example, an information collector can monitor user input and organize information into structured data for better future utilization.

Sometimes, a conversation is aimed at solving certain tasks during or after the conversation. When a conversation achieves the predefined goal, the system can call more complex programs to meet needs. Some tool API calls can also be added into the action list for execution, which means the action list is the interface of `DiagGPT` and can provide many plugins to enrich the functions of this AI system. This also means that it can expand goal-oriented dialogue to task-oriented dialogue by triggering the API of certain tasks.

4 Experiments

4.1 Setups

We conduct experiments to demonstrate the performance of `DiagGPT`. We first present a complete dialogue in the medical consulting process to show qualitative results of our AI system. This is then followed by a case study of automatic topic management, which details the changes in the topic stack during the dialogue process.

In the implementation of our AI system, we employed *gpt-4* as the base LLM, leveraging its strong understanding and reasoning abilities to achieve ideal results. We set the decoding temperature of all LLMs in our AI system to 0 to ensure more stable task execution. We provide detailed prompts in our AI system. The main prompts of the AI agent are shown in Figure 9, while Figure 10 displays the prompts for all actions in the action list. In these prompts, *{variable}* in blue indicates that the slot needs to be filled with the corresponding variable text. Once filled, these prompts can be fed into the LLMs to generate responses. Some of these slots facilitate information interaction between AI

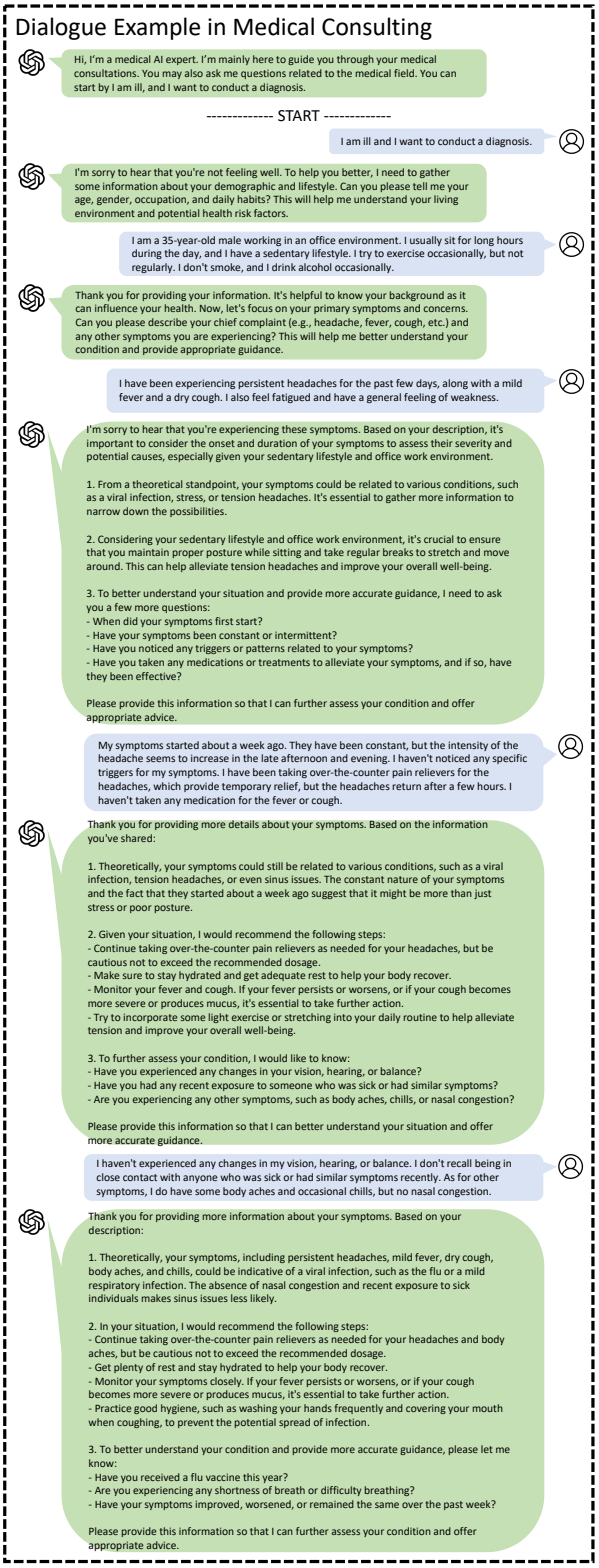

Figure 6: A dialogue example in the medical consulting scenario. The AI system acts as a real doctor in our daily life.

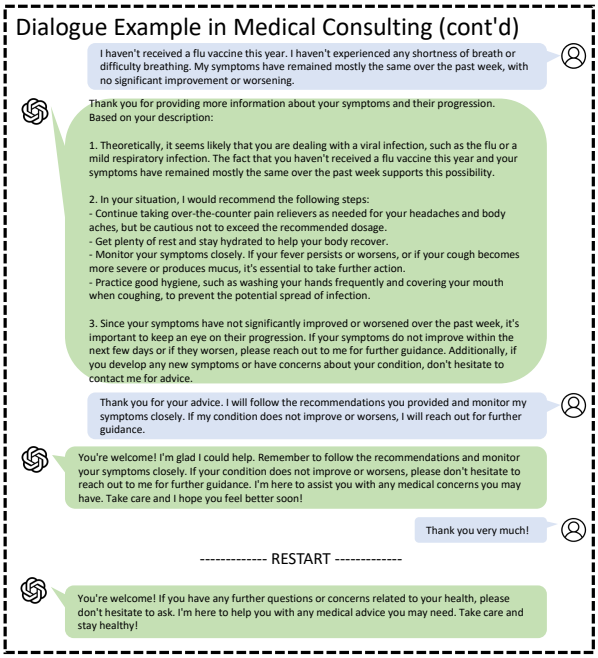

Figure 7: The continued dialogue example in the medical consulting scenario.

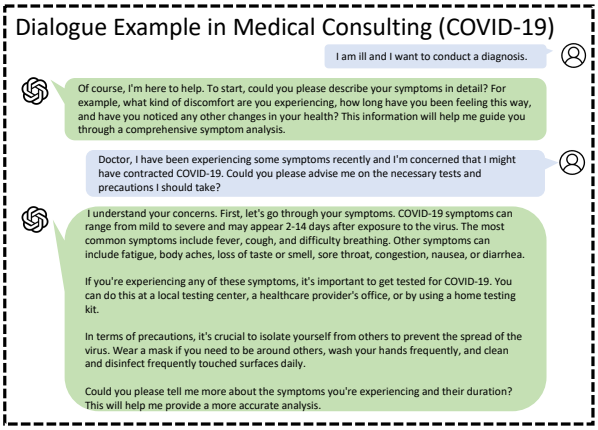

Figure 8: A dialogue example in the medical consulting scenario when user ask some questions about COVID-19.

systems.

We do not and cannot evaluate our system quantitatively. The test data for user queries is generated by another *gpt-4* model, which we design some prompts to guide. We do not evaluate our system using any goal-oriented dialogue datasets, like DialoGLUE(Yadav et al., 2019; Moghe et al., 2023), as DiagGPT is an open system that largely depends on AI consciousness combined with prompt instruction for output. We do not need to train our systems like fine-tuning models. Instead, we only need to provide a topic checklist and improve the instruction of the AI system to achieve ideal results, which are difficult to compare using quantitative metrics. However, qualitative experiments can fully show the ability and excellent performance of DiagGPT.

4.2 Qualitative Results

Figure 6 and Figure 7 present a complete dialogue demonstration in the medical consulting process. This is a medical diagnosis where the task is to help the patient identify the cause and give advice. Medical consulting is not a pure open-end question-answering. Because patients usually lack extensive medical knowledge, they rely on doctors to instruct them to give their personal information. Therefore, users can experience a real doctor, not a dull medical question-answering machine. The user acts as a patient, while the AI emulates a doctor, initially collecting information and gradually providing advice to the patient. The main dialogue development follows the checklist: *Basic information*, *Chief complaint*, *Duration of symptoms*, *Severity of symptoms*, which are also predefined topics.

Due to space constraints in this paper, we cannot present more dialogue examples. However, this demonstration has already showcased the robust conversational ability of the DiagGPT, which can actively ask questions and guide the user to the final goal of the task, thereby achieving goal-oriented dialogue. It simulates many real consulting scenarios. Other chat models, such as ChatGPT, cannot achieve this performance. They usually just answer user questions and find it challenging to complete specific goals, even with elaborated prompts.

4.3 Case Study of Automatic Topic Management

The core capability of DiagGPT is to automatically manage topics throughout the dialogue, which is the function and primary responsibility of *Topic Manager*. Figure 6 and Figure 7 illustrate the primary checklist progression. The topics from the checklist are retrieved and discussed sequentially, demonstrating the action of *finishing the current topic*. When the conversation reaches the severity of symptoms, we observe that the dialogue topic remains here in several rounds of dialogue, allowing the system to have time on understanding the user's conditions. This mirrors real-world scenarios where users do not provide enough information for the doctor. The action of *creating a new topic* is shown in Figure 8. Here, the user actively consults the system with some information about COVID-

19 to check symptoms. We observe that the AI generates a new topic about COVID-19 and discusses this, rather than rigidly following the checklist. These results all demonstrate the effectiveness of *Topic Manager*. These case studies fully embody the AI's flexible understanding ability, demonstrating its adept handling of different situations, closely mirroring real-world interactions.

5 Conclusion

In this paper, we propose DiagGPT, a multi-agent and collaborative AI system designed to complete goal-oriented dialogue tasks. The principle of our system is to leverage the strong understanding and reasoning capabilities of Large Language Models to design an AI agent that can automatically manage topics and track dialogue state. Therefore, our system can accurately understand users' intentions and help them to complete some specific tasks. DiagGPT demonstrates the unlimited potential of LLMs in more complex dialogue scenarios, such as goal-oriented dialogues, benefiting society by applying AI in various scenarios.

As previously mentioned, DiagGPT has ample room for extending its functionality. We aim to explore how to better serve users by combining the robust capabilities of LLMs. Moreover, we believe that the construction of LLM-based multi-agent systems signifies the future of AI development. We hope that the design of our system can inspire the development of more sophisticated AI applications and pave the way for LLMs towards more advanced AI systems.

Limitations

Hallucination. Hallucination is a major concern when it comes to the response of LLMs. Some individuals argue that LLMs cannot be utilized in serious scenarios such as healthcare or legal domains due to hallucinations. Besides, we acknowledge that there are some harmful responses from LLMs. On one hand, this paper does not primarily focus on mitigating hallucination. On the other hand, significant progress has been made in reducing hallucination of LLMs through various other studies. Thus, the application of this conversational system still holds immense potential in real-life situations.

Cost and Efficiency. DiagGPT involves multiple LLMs. In just one round of dialogue with a single user query, all of these LLMs need to run with elaborated prompts, which is quite costly. Compared to simpler dialogue systems that involve just one LLM interacting with users, DiagGPT requires internal interactions among AI agents, thus taking more time to provide user feedback. Furthermore, AI agent in our system requires strong understanding and reasoning abilities, necessitating a robust and large-scale AI to maintain these capabilities. This also increases the overall system cost. However, we believe that with the future development of AI infrastructure, these issues could be mitigated.

Stability. The performance of DiagGPT is not as stable as some rule-based or fine-tuned dialogue models. The main issue arises when the *Topic Manager* decides the direction of dialogue development. It requires a strong understanding and reasoning ability from the AI, or else it may lead to system instability. Additionally, for every different applied scenario, meticulous and detailed prompt adjustments of the AI system are needed. Given the risk of LLMs' output, some post-processing of responses is also required. Nevertheless, DiagGPT maintains unlimited potential for dialogue systems. With stronger AI in the future, dialogue systems could improve significantly and become more human-like.

Limited Experiments. We are unable to conduct quantitative experiments to demonstrate the performance of DiagGPT in the most direct manner, as we previously mentioned due to its inherent characteristics. Additionally, the scope of the results obtained from quantitative experiments is limited. Although we provide only one usage example in medical consulting, our system can also exhibit similarly impressive performance in legal and numerous other scenarios because of the extensive knowledge from LLMs. The basic operational processes in different applications are the same. Moreover, through careful observation of the responses generated by DiagGPT, we can discern that each module effectively fulfills its role, thus showcasing its overall strong performance.

References

Vevake Balaraman, Seyedmostafa Sheikhalishahi, and Bernardo Magnini. 2021. Recent neural methods on dialogue state tracking for task-oriented dialogue sys-

tems: A survey. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 239–251, Singapore and Online. Association for Computational Linguistics.

Namo Bang, Jeehyun Lee, and Myoung-Wan Koo. 2023. Task-optimized adapters for an end-to-end task-oriented dialogue system. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7355–7369, Toronto, Canada. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways.

Yue Feng, Yunlong Jiao, Animesh Prasad, Nikolaos Aletras, Emine Yilmaz, and Gabriella Kazai. 2023. Schema-guided user satisfaction modeling for task-oriented dialogues. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2079–2091, Toronto, Canada. Association for Computational Linguistics.

Vojtěch Hudeček and Ondřej Dušek. 2023. Are llms all you need for task-oriented dialogue?

Pan Liang, Danwei Ye, Zihao Zhu, Yunchao Wang, Wang Xia, Ronghua Liang, and Guodao Sun. 2023. C5: Towards better conversation comprehension and contextual continuity for chatgpt.

Yajiao Liu, Xin Jiang, Yichun Yin, Yasheng Wang, Fei Mi, Qun Liu, Xiang Wan, and Benyou Wang. 2023. One cannot stand for everyone! leveraging multiple user simulators to train task-oriented dialogue systems. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1–21, Toronto, Canada. Association for Computational Linguistics.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Nikita Moghe, Evgeniia Razumovskaia, Liane Guillou, Ivan Vulić, Anna Korhonen, and Alexandra Birch. 2023. Multi3NLU++: A multilingual, multi-intent, multi-domain dataset for natural language understanding in task-oriented dialogue. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 3732–3755, Toronto, Canada. Association for Computational Linguistics.

OpenAI. 2023. Gpt-4 technical report.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2023. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022a. Emergent abilities of large language models.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022b. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2017. A network-based end-to-end trainable task-oriented dialogue system. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 438–449, Valencia, Spain. Association for Computational Linguistics.

Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019. Transferable multi-domain state generator for task-oriented dialogue systems. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 808–819, Florence, Italy. Association for Computational Linguistics.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2022. An explanation of in-context learning as implicit bayesian inference. In *International Conference on Learning Representations*.

Deshraj Yadav, Rishabh Jain, Harsh Agrawal, Prithvijit Chattopadhyay, Taranjeet Singh, Akash Jain, Shiv Baran Singh, Stefan Lee, and Dhruv Batra. 2019. Evalai: Towards better evaluation systems for ai agents.

Qiang Zhang, Jason Naradowsky, and Yusuke Miyao. 2023. Ask an expert: Leveraging language models to improve strategic reasoning in goal-oriented dialogue models.

A Prompt Design

The content of prompts for some LLMs in `DiagGPT` are shown in Figure 9 and Figure 10.

Prompt of Chat Agent

```
You are a very good and famous doctor and AI medical expert who works for patients. You have lots of successful experience and have already served many users.
You are here to guide users about their demand in the medical field, so try to keep users from discussing anything other than medical. Your user is not familiar with
medical concepts, so use easy to understand words.
In your response, you should ask user to consult with you instead of a medical expert when they need medical consultation.
You have been trained to be a very good medical expert. Never ask user to consult with a medical expert!
Never ask user to consult with a medical expert!
Never ask user to consult with a medical expert!

Your current dialogue topic is: {current_topic}
NOTE: you must always focus on this topic in this round of dialogue!
If current topic is asking the user, you should ask the user about it concretely and directly.
If current topic is answering user:', your answer should contain three parts: 1. General answer from theoretical analysis, 2. Specific suggestions to users based on
user scenarios, 3. Ask the user for some information to clarify the question, so as to further better answer the question.
If current topic is completing a goal, you should directly give a comprehensive and detailed answer based on the chat history to meet the final goal.

The overview of your current task in the entire dialogue is: {task_overview}
The final goal of your current task in the entire dialogue is: {final_goal}
NOTE: you should always try to lead dialogue topic development to this goal, avoid going off topic, and complete this goal finally.

###### General Knowledge START ######
(NOTE: these materials are from other scenes for reference, and should not be considered as part of user information)
{background_context}
###### General Knowledge END ######

###### User Knowledge START ######
(These information are provided by the user)
{user_context}
###### User Knowledge END ######

###### Chat History ######
{chat_history}

Current conversation:
User: {human_input}
AI (Medical Expert): (If users ask a question, remember to give concrete and specific suggestions based on user scenarios. Never ask users to consult with a medical
expert because you are already a very good and famous medical expert!)
```

Prompt of Topic Manager

```
Your objective is to manage dialogue topic in the dialogue between a AI medical expert and a user.
The dialogue topics are always about the medical field. If you can effectively manage topics, the AI medical expert can have a better dialogue with users
You now have a topic list which contains all existing topics in the entire dialogues in order, which are delimited by triple backticks: ```{topic_list}```.

In this topic list, topic are separated by semicolon (; ) in the topic list, and a topic includes the content in parentheses (())!
The current dialogue topic is the last topic in the topic list, which is {current_topic}.
In general, when you finish the current topic, the next dialogue topic is the second to last in the topic list.
In general, topic development usually follows the reverse order of the list, unless the user needs to create some new topics.

You need to manage dialogue topics as best as you can using the following tools:

{tool_description}

###### AI medical expert Chat History START ###### (you can consider previous chat history between the AI medical expert and the user)
{chat_history}
###### AI medical expert Chat History END ######

You must use the following format, including User Input, Thought, Action, Action Input, and Observation:

User Input: the input from the user
Thought: comment on what you want to do next
Action: the action to take, exactly one element of [{tool_names}]
Action Input: the input to the action (if you are using a tool without input, Action Input should be None)
Observation: the result of the action (STOP here)

###### STOP ###### (just think one round, after give Observation, you must STOP! STOP! STOP!)

Begin!

User Input: {human_input}
Thought: (HINT: focus on the last output of AI medical expert the current input of the user)
```

Prompt of Topic Enricher

```
Your objective is to enrich dialogue topics between a AI medical expert and a user. I will give you an original and simple topic, and you need to give me an enriched
topic based on the original one and my needs.
The new enriched topic will be used by a AI medical expert, which is also trained from ChatGPT, like you. This topic can be thought of as a prompt. The AI medical
expert need to first understand the new topic and then talk to users about this topic.
If you give a better topic to the AI medical expert, it can have a better dialogue with users, so craft the best possible topic (prompt) for my needs.

Make sure that the AI medical expert can understand it easily!
Your new topic needs to for AI medical experts to tell it what to do, not users!
Your new topic needs to for AI medical experts to tell it what to do, not users!
Your new topic needs to for AI medical experts to tell it what to do, not users!

You need to consider previous chat history with the user to detail and improve the original topic:
###### Chat History START ###### (NOTE: do not use chat history in your topic directly)
{chat_history}
###### Chat History END ######

Provide your new topic. Your new topic is limited to 120 words. Remember your new topic needs to for AI medical experts to tell it what to do, not users!

Begin!

Original Topic: {original_topic}
New Topic:
```

Figure 9: The prompts of *Chat Agent*, *Topic Manager*, *Topic Enricher*. We have included instructions to guide the AI in becoming a knowledgeable medical expert, making it applicable in medical dialogue scenarios. These instructions can be modified to suit other scenarios.

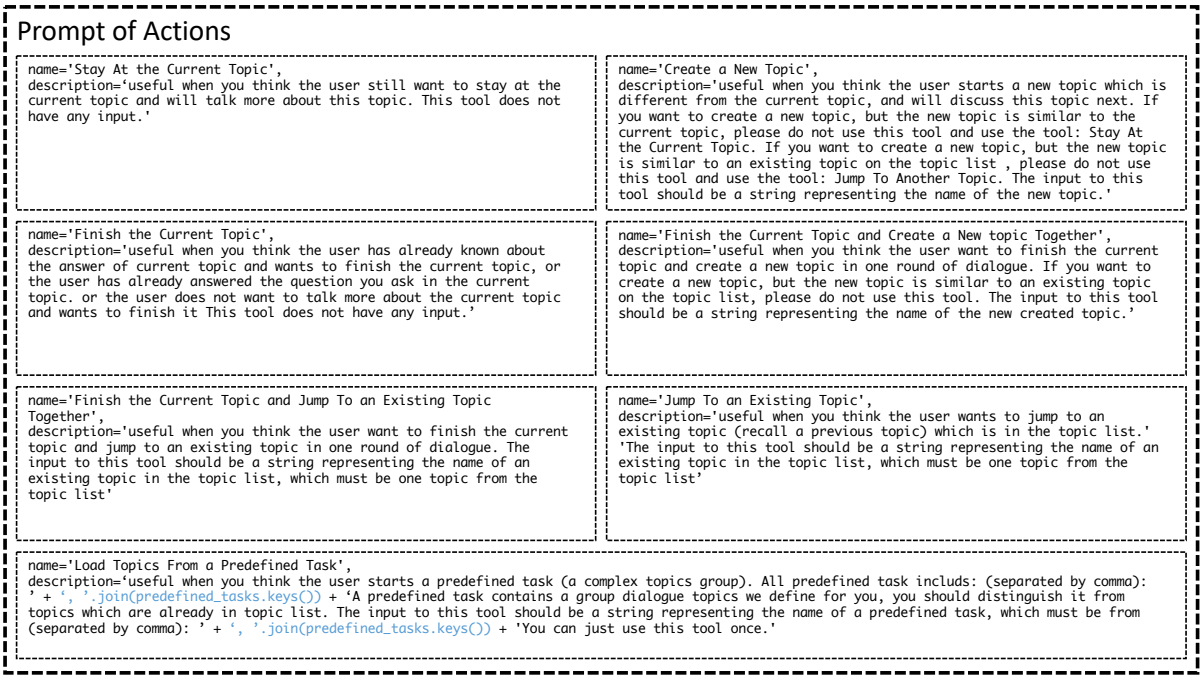

Figure 10: The prompts for different actions are used to define specific program functions that correspond to their respective actions and instruct when to execute them.