

TONGJI UNIVERSITY

课题名称 　Technical Report on Multi-Model Intelligent Dialogue Systems

副 标 题 　　　　用户交互技术作业2

学 院 　　　 计算机科学与技术学院

专　　业 　　　　　 软件工程

学生姓名 　　　 关镜文　　黄弋涵

学　　号 　　　 2352744　 2351271

指导教师 　　　　　　 沈莹

日　　期 　　　　 2025年4月20日

**1. Differences in API Calls of Different Models**

In the system integration phase, there are significant differences in the API calls between DeepSeek and OpenAI. From the perspective of SDK configuration, DeepSeek requires an explicit specification of the endpoint URL and configuration of the service address through the base\_url parameter, while OpenAI uses the official endpoint by default, requiring only the API key for initialization. This difference reflects the varying strategies of different service providers in API access design, highlighting their emphasis on user experience and developer-friendliness.

Regarding model parameter configuration, DeepSeek adopts a preset model name deepseek-chat, with its system prompt function being in a pre-configured mode, allowing developers to use the default settings without additional configuration. This design simplifies the development process and lowers the entry barrier. In contrast, OpenAI's gpt-4o-mini model offers more flexible options for configuring system prompts, enabling developers to customize the system role based on specific scenario requirements, providing greater freedom and creative space. In terms of error handling mechanisms, DeepSeek implements error tracing through a custom label [DeepSeek Error], while OpenAI uses an independent prefix identification system. This design difference directly impacts the exception handling processes of the systems, potentially leading to differences in debugging convenience for developers.

At the architectural implementation level, DeepSeek employs an explicit dialogue history tracking mechanism, maintaining contextual relationships through an independent state management module. This design ensures the coherence of dialogues and the accuracy of context. In contrast, OpenAI adopts a simplified prompt processing flow, relying on historical dialogue records within the request message to maintain context continuity. These architectural differences result in unique technical characteristics in memory management and request load handling, affecting their performance in high-concurrency scenarios.

**2. User Interface Design Approach**

**(1)Core Interaction Framework**

The user interface of this system adopts a split-screen interaction architecture, with the upper part serving as the main operation area and the lower part as the output results area, creating a clear and intuitive visual flow. This design not only enhances user operational efficiency but also makes information presentation more straightforward. The top navigation bar integrates a model selection control, allowing users to switch between "DeepSeek" and "OpenAI" with a toggle button, providing clear functionality and easy understanding, enabling users to quickly select the desired dialogue model based on their needs.

The input area utilizes a floating toolbar design, with clearly defined button functionalities, allowing users to operate conveniently. The intuitive design of the "Send" and "Clear" buttons enables users to respond quickly during input, enhancing the overall user experience.

**(2)Dynamic Dialogue Panel**

The dialogue area implements layered rendering technology, effectively separating real-time dialogue flow from historical dialogue, with clear and distinct markings. Historical dialogues display the current model's responses with color-coded borders (orange for DeepSeek and blue for OpenAI), which enhances usability by allowing users to quickly identify responses from different models.

The historical dialogue recovery system is implemented through a "Continue Conversation" button, with each record marked by a model source icon and timestamp, displaying the user's input and the model's reply at that time, facilitating users' ability to trace and continue previous dialogues. This design not only improves user interaction but also enhances the coherence of conversations and the understanding of context.

**3. Impact of Large Model Parameters**

The generation parameter system has a decisive impact on output quality. The temperature parameter (Temperature) serves as a core control factor, achieving a smooth transition from deterministic output to creative expression within the range of 0.2 to 0.8. Specifically, when set to 0.2, it is suitable for scenarios requiring high accuracy, such as code generation, while raising it to 0.8 can stimulate the literary expression needed for poetry creation, showcasing higher creativity. The top-p sampling parameter dynamically adjusts the probability space for token selection; at a setting of 0.9, it allows the model to focus on the top 10% of high-probability candidate words, effectively balancing response quality and diversity.

In terms of response length control, the maximum token count parameter directly affects the level of detail in the generated content. A setting of 500 tokens is suitable for generating long texts, such as thesis outlines, while a setting of 100 tokens optimizes response efficiency in Q&A scenarios, ensuring that users can quickly obtain the information they need. The frequency penalty mechanism, set with a coefficient of 1.5, can reduce the repetition rate of technical terms in documents by 40%, significantly enhancing content readability. The collaborative effect of these parameters constitutes a complete generation control matrix, ensuring the model's adaptability and flexibility across various scenarios.

**4. Prompt Engineering Strategies**

In human-computer dialogue, the differences in prompt words significantly affect the output of large models. This impact is primarily reflected in several aspects.

First and foremost, the clarity of the prompt is crucial. When users provide specific prompts, the model can more clearly understand their intentions, leading to more relevant responses. For instance, asking, “Please provide a brief summary of climate change” is more likely to yield targeted answers than simply saying, “Tell me something.” In contrast, vague or open-ended prompts may result in the model generating less relevant or overly broad content.

Additionally, the provision of context greatly influences the model's performance. When users include contextual information in their prompts, the model can better grasp the background of the question. For example, a prompt like, “List several key issues when discussing the ethics of artificial intelligence” helps the model focus on a specific topic, whereas a prompt lacking context may lead to misunderstandings.

Moreover, format requirements are equally important. When users wish to receive information in a specific format, clearly stating this can assist the model in generating outputs that meet expectations. For example, a user might request, “Please list five benefits of healthy eating,” allowing the model to organize its response accordingly. Conversely, if no format is specified, the output may not align with the user’s preferences.

The indication of tone and emotion also affects the model’s language style and voice. For instance, users can request the model to “explain in a humorous way” or “maintain a formal tone,” and these emotional cues can make the output more aligned with the user's needs. Without such emotional indications, the model might produce responses that are neutral or not in line with user expectations.

Finally, the depth of knowledge required can also influence the model's response. When users request in-depth analysis or detailed explanations, the model is likely to provide more complex information. For example, asking, “Explain the basic principles of quantum computing and its applications” will elicit a more thorough response, whereas simpler questions might only receive superficial answers.

In summary, designing effective prompts is key to engaging in efficient dialogue with large models. By optimizing aspects such as clarity, context, format, tone, and knowledge depth, users can obtain more accurate, useful, and expected responses, thereby enhancing the quality and efficiency of human-computer interaction.

So we recommend the principles of scenario adaptation for prompts as follows.

In everyday dialogue scenarios, natural language expression modes are used, such as open-ended instructions like "Explain quantum physics in simple language," to ensure that users can easily understand and utilize the system.

When faced with specialized questions, structured templates are employed, constructing the prompt framework through role definitions, task breakdowns, and input specifications. A typical example is “You are a senior front-end developer who needs to complete the development of a web platform, currently having completed the homepage and login page displays,” gradually increasing detail requirements in multi-turn dialogues.

The four key dialogue techniques are as follows: role specification, which guides the model into a professional discourse system through identity anchoring (e.g., "senior software architect"). Context anchoring, which establishes semantic associations using key information from dialogue history. Response formatting requirements, which can standardize output structure using Markdown syntax. Iterative optimization, which continuously adds constraints and supports dynamic adjustments based on conditions.b

The combined application of these strategies can enhance prompt efficiency and improve model response accuracy, thereby enriching user experience.

**5. Conclusion**

The multi-model intelligent dialogue system exhibits significant differences and characteristics in API calls, user interface design, model parameter impacts, and prompt engineering. By analyzing the different strategies of DeepSeek and OpenAI, we can better understand their respective advantages and limitations. In practical applications, selecting the appropriate model and configuring parameters will directly affect system performance and user experience. Furthermore, a well-designed user interface and effective prompt strategies can significantly enhance interaction fluidity and response accuracy. In the future, with continuous technological advancements and diverse user needs, intelligent dialogue systems will evolve towards greater flexibility and efficiency, providing users with higher-quality services and experiences.

**References**

1. Rossouw, A., Smuts, H. (2023). Key Principles Pertinent to User Experience Design for Conversational User Interfaces: A Conceptual Learning Model. In: Huang, YM., Rocha, T. (eds) Innovative Technologies and Learning. ICITL 2023. Lecture Notes in Computer Science, vol 14099. Springer, Cham. <https://doi.org/10.1007/978-3-031-40113-8_17>
2. Isabel Kathleen Fornell Haugeland, Asbjørn Følstad, Cameron Taylor, Cato Alexander Bjørkli. (2022). Understanding the user experience of customer service chatbots: An experimental study of chatbot interaction design. https://doi.org/10.1016/j.ijhcs.2022.102788.
3. Rheu, M. et al. (2020) ‘Systematic Review: Trust-Building Factors and Implications for Conversational Agent Design’, International Journal of Human–Computer Interaction, 37(1), pp. 81–96. doi: 10.1080/10447318.2020.1807710.
4. Chaves, A. P. and Gerosa, M. A. (2020) ‘How Should My Chatbot Interact? A Survey on Social Characteristics in Human–Chatbot Interaction Design’, International Journal of Human–Computer Interaction, 37(8), pp. 729–758. doi: 10.1080/10447318.2020.1841438.
5. Standardization, IOf. "Ergonomics of human-system interaction." Human-centred design for interactive systems. ISO (2019).
6. Fadhil, A.: Domain specific design patterns: designing for conversational user interfaces. [arXiv:1802.09055](http://arxiv.org/abs/1802.09055) (2018)