

DUMA: a Dual-Mind Conversational Agent with Fast and Slow Thinking

Xiaoyu Tian, Liangyu Chen, Na Liu, Yaxuan Liu,
Wei Zou, Kaijiang Chen, Ming Cui

Beike Inc., Beijing, China

{tianxiaoyu011, chenliangyu003, liuna013, liuyaxuan002,
zouwei026, chenkaijiang001, cuiming001}@ke.com

Abstract

Inspired by the dual-process theory of human cognition, we introduce DUMA, a novel conversational agent framework that embodies a dual-mind mechanism through the utilization of two generative Large Language Models (LLMs) dedicated to fast and slow thinking respectively. The fast thinking model serves as the primary interface for external interactions and initial response generation, evaluating the necessity for engaging the slow thinking model based on the complexity of the complete response. When invoked, the slow thinking model takes over the conversation, engaging in meticulous planning, reasoning, and tool utilization to provide a well-analyzed response. This dual-mind configuration allows for a seamless transition between intuitive responses and deliberate problem-solving processes based on the situation. We have constructed a conversational agent to handle online inquiries in the real estate industry. The experiment proves that our method balances effectiveness and efficiency, and has a significant improvement compared to the baseline.

1 Introduction

In the era of rapid progress of LLMs(OpenAI, 2023a; OpenAI, 2023b), creating conversational agents(Wang et al., 2023a; Xi et al., 2023; Weng, 2023) that can emulate human-like interactions is both a challenge and an aspiration. Drawing inspiration from the dual-process theory(Daniel, 2017) of human cognition, which proposes two distinct cognitive processes—a fast, intuitive one and a slower, analytical one. we introduce DUMA, a novel conversational agent framework(Zalta et al.,

1995). While there have been efforts like Swift-Sage(Lin et al., 2023) that delve into integrating fast and slow thinking processes in AI agents, DUMA stands out by prioritizing conversational scenarios.

DUMA, symbolizing a Dual-Mind Conversational Agent, merges two generative Large Language Models (LLMs) to accommodate distinct cognitive processes: fast and slow thinking. The Fast Mind module of DUMA stands out for its agility and efficiency, readily addressing straightforward scenarios. However, its nimbleness might falter in more intricate situations. Conversely, the Slow Mind takes its time, operating at a deliberate pace which might seem less efficient. Yet, this deliberation equips it to grapple with complex challenges, particularly those demanding in-depth reasoning or the invocation of external tools. Together, these two minds allow DUMA to deliver a balanced conversational experience, oscillating smoothly between immediate replies and profound problem-solving.

DUMA’s dual-mind structure is reminiscent of human cognitive processes, differentiating between the Fast Mind and the Slow Mind. For routine queries, the Fast Mind takes the lead with immediate responses. However, when complex queries arise that demand deeper analysis, like mathematical or logical challenges, it calls upon the Slow Mind. Unlike its counterpart, the Slow Mind doesn’t interact directly with users. Instead, it delves into the problem using a methodology inspired by ReAct(Yao et al., 2023), often calling upon external tools for assistance. Once its comprehensive analysis is done, the insights are relayed back to the Fast Mind, which crafts the response. Importantly, these insights are archived in the Fast Mind’s "Memory Area", ensuring efficiency in future related dialogues, epitomizing DUMA’s blend of agility and depth.

We’ve conducted experiments in Chinese real

estate online communication scenarios, confirming the efficacy of our approach. Although our experiments were specific to the Chinese context and real estate domain, we believe that the methodology behind DUMA possesses a broader applicability.

Our main contributions in this paper are two folds:

- Introducing DUMA, a novel conversational agent framework built upon the dual-process theory, integrating two LLMs for fast and slow cognitive processes.
- Demonstrating the application and efficacy of DUMA in real-world scenarios, specifically in the real estate industry, where it showcases significant improvements over baseline models.

2 Methodology

In this section, we will first introduce the overall structure of DUMA, then set forth the design and operation process of Fast Mind and Slow Mind respectively, and finally elaborate the internal and external interaction of DUMA.

2.1 DUMA Overall Structure

The thinking center of DUMA contains two minds, Fast Mind (defined as $Mind_{Fast}$) and Slow Mind (defined as $Mind_{Slow}$), as shown in the Figure 1.

As discussed above, simple questions in general conversations do not require in-depth thinking. $Mind_{Fast}$ is responsible for quickly thinking, replying to those simple questions, and directly interacting with human. For those difficult questions, such as mathematics, complex reasoning, etc., after $Mind_{Fast}$ perceives the complexity of the question, it will send a signal to $Mind_{Slow}$, thereby triggering DUMA's deep thinking mechanism.

$Mind_{Slow}$ will not directly interact with human. After deep thinking, $Mind_{Slow}$ will transmit the results and data obtained after calling the tool back to $Mind_{Fast}$. The current turn of dialogue will be generated by $Mind_{Fast}$ based on the inference results of $Mind_{Slow}$. Finally, DUMA will return the result through $Mind_{Fast}$.

2.2 Fast Mind

For each turn of dialogue, $Mind_{Fast}$ may have two input sources, human utterance or $Mind_{Slow}$

thinking results. $Mind_{Fast}$ infers each turn of responses based on historical dialogue and current input.

Formally, for time step t , assuming the human utterance is Q_t , the $Mind_{Slow}$ thinking result is S_t , the input will be processed into the following format:

$$I_t^{Fast} = \begin{cases} User[Q_t], & \text{Human Input} \\ SlowMind[S_t], & Mind_{Slow} \text{ Input} \end{cases} \quad (1)$$

According to the previous dialogue, $Mind_{Fast}$ quickly thinking and generating answer. The results generated by $Mind_{Fast}$ can indicate whether $Mind_{Slow}$ needs to be triggered and what needs to be replied. $Mind_{Fast}$'s response in round t is defined as O_t :

$$O_t^{Fast} = \begin{matrix} Invoke[V], V \in \{True, False\} \\ Response[Mind_{Fast} \text{ output}_t] \end{matrix} \quad (2)$$

When the value in "Invoke" is True, $Mind_{Slow}$ will be awakened. At this time, DUMA will think deeply and thoroughly. The content in "Response" is the reply to the t -th turn of conversation.

Analogous to human, $Mind_{Fast}$ conducts multi-turn of dialogue with individuals. For the current turn of response, $Mind_{Fast}$ needs to "consider" the context of past dialogue, therefore, we need to process historical conversation into multi-turn format that the model can understand:

$$Context_t = \underbrace{M_b I_0 M_e O_0}_{conv_0} \underbrace{M_b I_1 M_e I_1}_{conv_1} \dots \underbrace{M_b I_{t-1} M_e O_{t-1}}_{conv_{t-1}} M_b I_t M_e \quad (3)$$

M_b and M_e are the pattern of the LLM's multi-turn dialogue¹. $Mind_{Fast}$ performs quick thinking based on $Context_t$, and generates the result O_t :

$$O_t^{Fast} = Mind_{Fast}(Context_t) \quad (4)$$

2.3 Slow Mind

When $Mind_{Slow}$ is awakened, just like human, $Mind_{Slow}$ will first retrospect the past dialogue.

¹The pattern of multi-turn of dialogue may be different for different open source LLMs.

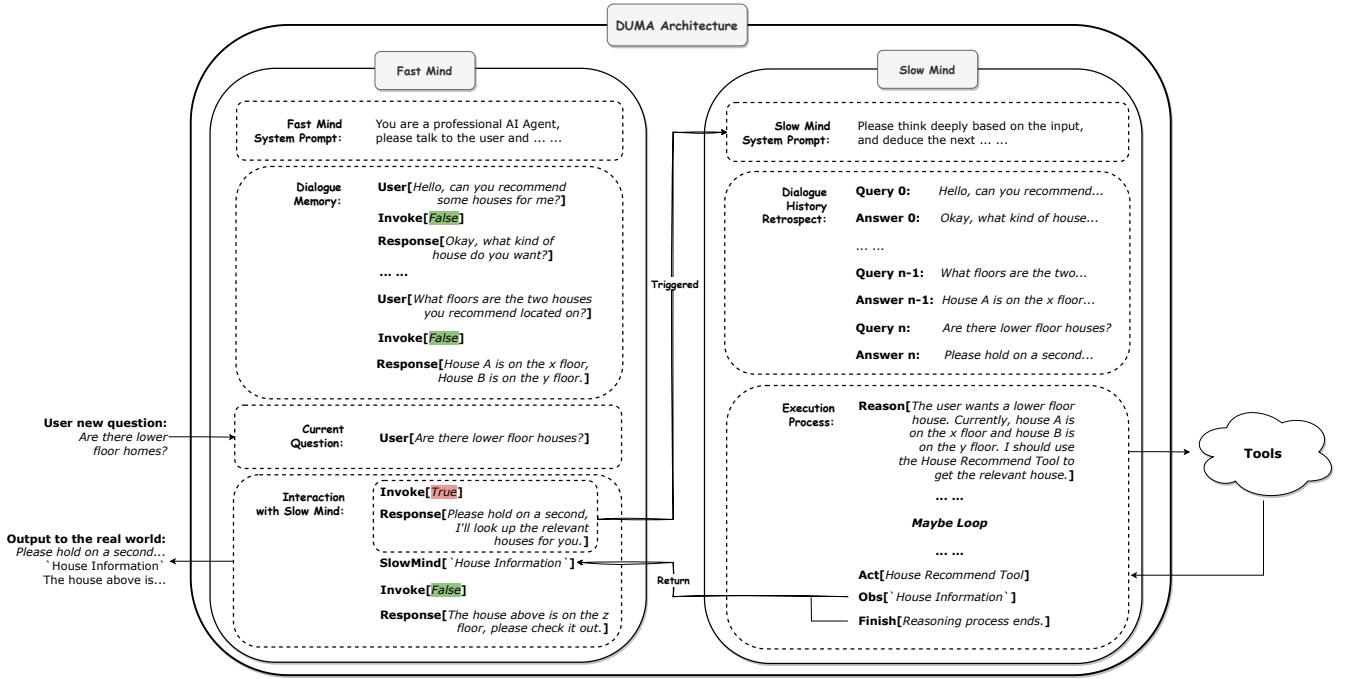


Figure 1: Upon receiving a user question, both the question and Dialogue Memory are fed into Fast Mind. If deep analysis is deemed necessary, Slow Mind is activated. Slow Mind reviews historical dialogues, conducts reasoning, takes action, and observes. The results are relayed back to Fast Mind, which crafts a response based on Slow Mind’s feedback, delivering it to the real-world context.

Assume that the $Mind_{Fast}$ response in round t is A_t , and the past dialogue will be structured by $Mind_{Fast}$ in the following format:

$$Dialogue_t^{his} = \begin{cases} Query\ 0 : Q_0 \\ Answer\ 0 : A_0 \\ \dots \\ Query\ t - 1 : Q_{t-1} \\ Answer\ t - 1 : A_{t-1} \\ Query\ t : Q_t \end{cases} \quad (5)$$

Upon encapsulation of $Dialogue_t^{his}$ by $Mind_{Fast}$, the content is subsequently conveyed to $Mind_{Slow}$ to denote the historical dialogue exchanged between $Mind_{Fast}$ and individuals. Subsequent to this, $Mind_{Slow}$ engages in reasoning, action, and observation:

$$O_t^{Slow} = Mind_{Slow}(Dialogue_t^{his}) \quad (6)$$

Inspired by ReAct(Yao et al., 2023), we divide the $Mind_{Slow}$ thinking process into four parts (Reason, Act, Obs, Finish), as shown in the Figure 1.

When encountering complex problems, $Mind_{Slow}$ may reason in multiple steps. For

example, when external tool support is needed, $Mind_{Slow}$ calls the tool through "Act" and observes the results of the tool through "Obs". Then the current review of the t -th turn of dialogue, all past reasoning, action and observation will be encapsulated into a new chain of thought and input into $Mind_{Slow}$ which will judge whether the next step of in-depth thinking is needed. If necessary, the process stated above will be looped, otherwise will be finished.

The value in "Obs" or "Finish" is the final reasoning result of $Mind_{Slow}$, which is encapsulated into the Formula 1 ($Mind_{Slow}$ Input) and then input into $Mind_{Fast}$.

2.4 DUMA: Fast Mind + Slow Mind

$Mind_{Fast}$ interacts with the real world to obtain the latest question and interacts with the $Mind_{Slow}$ (if necessary) to obtain in-depth thinking results. "Dialogue Memory" shown in Figure 1 consists of two parts: the interaction between $Mind_{Fast}$ and the real world, and the results returned by $Mind_{Slow}$ (including observation, inference results, etc.). For the same situation that needs to trigger $Mind_{Slow}$, DUMA only needs to think deeply once, and the results are saved in the "Dialogue Memory" of $Mind_{Fast}$ in the

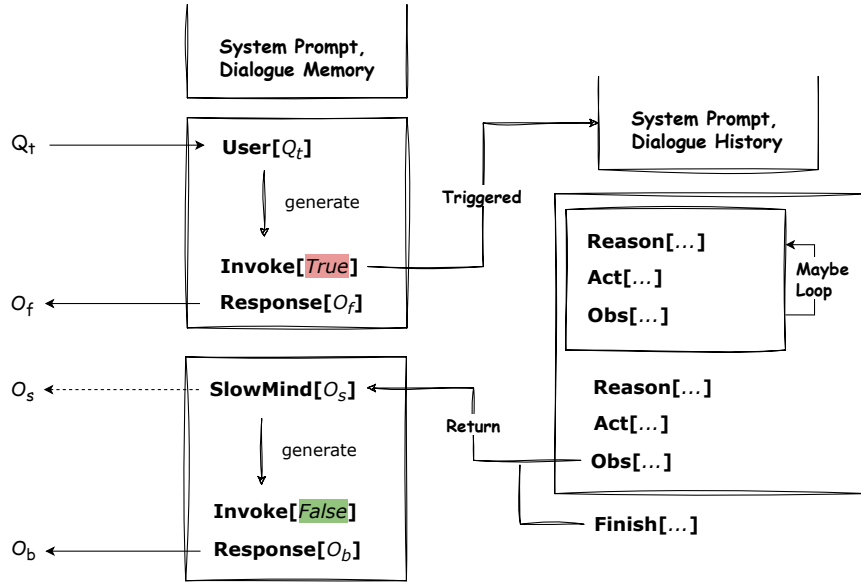


Figure 2: Fast Mind produces the current result based on Dialogue Memory and Q_t . When Invoke is set to True, the deep thinking mechanism of Slow Mind is triggered. Slow Mind then returns its observation and reasoning results to Fast Mind. Whether O_s is relayed to the real world is determined by the system. Ultimately, O_f , O_s , and O_b are returned to the real world as the complete results of the current round.

form of dialogue. When asked questions related to memory, $Mind_{Fast}$ quickly generates and responds based on the interactive memory with the real world and $Mind_{Slow}$, thereby improving the conversation efficiency of the entire agent.

As shown in the Figure 2, when DUMA receives a new question, the system prompt, historical data and current questions will be spliced and input into $Mind_{Fast}$ for generating. When $Mind_{Slow}$'s in-depth thinking is triggered, system prompt and dialogue reviews (described in Section 2.3) are spliced and input into $Mind_{Slow}$ for reasoning, actions, and observations. This process may be looped, and $Mind_{Slow}$ decides when to end on its own. After $Mind_{Slow}$'s thinking is completed, the information in "Obs" is returned to $Mind_{Fast}$. $Mind_{Fast}$ makes inference responses based on the results of $Mind_{Slow}$, and finally returns O_f and O_b to the real world. The specific application decides whether O_s needs to be exposed.

3 Experiments

We can employ LLMs to construct an agent grounded on the DUMA architecture without training. However, given the complexity of online real estate conversation scenarios, Fast Mind employs Baichuan-13B-Chat(Technology, 2023) as its foundational model for training. Simultaneously, considering that Slow Mind might necessari-

tate multi-step reasoning, ChatGLM2-6B(Zeng et al., 2022; Du et al., 2022) is utilized as the base model for training to enhance the agent's performance and expedite the inference efficiency.

3.1 Data Collection

3.1.1 Fast Mind

Upon training with online real estate dialogue datasets, the model assimilates the conversational style and tone characteristic of actual individuals. However, we concurrently identify a concerning issue related to hallucination, such as consistency discrepancies, notably when the Fast Mind assimilates inputs from the Slow Mind, leading to the generation of erroneous content, as well as fabricating facts problems. To enhance the model's factuality, we implemented a two-phase SFT process: Dialogue Training followed by Factuality Enhancement.

SFT Stage I - Dialogue Training We obtain original 1.3M dialogue data from online dialogue logs and established a data processing pipeline, which mainly includes data desensitization, rule based filtering, logic optimization, intent balancing, etc. Finally, we collect 66349 multi-turn dialogues. Before training, the data will be processed into the model's multi-turn dialogue format. The first stage SFT data uses complete dialogue.

SFT Stage II - Factuality Enhancement To

Stage I Data		Stage II Data			
[... Q ₀ ...]	✗	[... Q ₀ ...]	✗	[... Q ₀ ...]	✗
[... A ₀ ...]	✓	[... A ₀ ...]	✓	[... A ₀ ...]	✗
⋮		⋮		⋮	
[... Q _t ...]	✗	[... Q _t ...]	✗	[... Q _t ...]	✗
[... A _t ...]	✓	[... A _t ...]	✓	[... A _t ...]	✓
[... Q _{t+1} ...]	✗	[... Q _{t+1} ...]	✗	[... Q _{t+1} ...]	✗
[... A _{t+1} ...]	✓	[... A _{t+1} ...]	✓	[... A _{t+1} ...]	✗
⋮		⋮		⋮	
[... Q _n ...]	✗	[... Q _n ...]	✗	[... Q _n ...]	✗
[... A _n ...]	✓	[... A _n ...]	✓	[... A _n ...]	✗

Data Ratio = 1:10

Figure 3: Differences in data construction and loss computation process between SFT Stage I (Dialogue Training) and Stage II (Factuality Enhancement). In Stage II, if factuality calibration is performed for A_t , subsequent dialogues are deleted to ensure dialogue coherence. In the Figure, "✓" indicates that the loss needs to be calculated, while "✗" indicates that the loss doesn't need to be calculated. For both Stage I and Stage II, loss is computed for each turn of the complete dialogue, while for Stage II's factuality calibration data, only the annotated replies require loss computation.

enhance the factuality of the model, we resampled 0.4M dialogues from online logs. After pipeline data processing (in order to ensure that the intent distribution is consistent with online, intent balancing is not performed), we acquire 65009 multi-turn dialogues. Based on the online probability distribution of intent and the likelihood of a given intent appearing at various positions within a dialogue, we randomly sampled 3,000 dialogues for factual calibration annotation. During annotation, only the response to the current turn of question required factual calibration. Given that the annotation might alter the logical coherence of subsequent dialogue, to ensure logical coherence, any content following the annotated dialogue is removed. The differences in data construction methods between Stage I and Stage II are illustrated in the Figure 3.

3.1.2 Slow Mind

We sample 0.5M raw dialogues from online logs. To facilitate better collaboration between Fast Mind and Slow Mind, the data underwent processing through a pipeline identical to Section 3.1.1 (excluding intent balancing), yielding 8,000 samples. To reduce the labeling cost, we label the data (has been desensitized) by GPT-4 pre-labeling and then manual correction. The method for data construction is elaborated in Section 2.3.

3.2 Experiments Setups

When training Fast Mind, the questions in the multi-turn dialogue of the first stage of SFT are not involved in the loss calculation. In the second stage of SFT, only the last turn of responses are involved in the loss calculation, and the gradients of the remaining questions and answers will be masked. To make the model better retain the dialogue logical coherence, during the second stage of training, we randomly sample 300 multi-turn dialogues from the first stage SFT training data (data ratio is 1:10) to conduct mixed training. The two-stage gradient calculation method is shown in the Figure 3.

We use the ChatGPT3.5 interface and adopt the ReAct(Yao et al., 2023) method as the baseline (denote as ChatGPT_{react}). When performing the first stage SFT of Fast Mind, we use Baichuan-13B-Chat as the basic model with learning rate of 1e-4, the second stage SFT uses the first stage checkpoint, and Slow Mind training with ChatGLM2-6B. The learning rate of the second stage SFT of Fast Mind and Slow Mind training are both 1e-5. Throughout all training procedures, the maximum length is 4096, training for 4 epochs, with a batch size of 32. We use a cosine LR schedule down to 10% of the original learning rate, with 3% warmup. All the models are trained

Table 1: The evaluation criteria details of the defined metrics.

Score	House Expertise	Tool Calling Ability	Industry Familiarity	Service Attitude	Demand Mining	Promote Invitation
0	All answered incorrectly.	All used incorrectly.	All answered incorrectly.	Not like human, or logically confusing.	No mining action was performed when needed, or all failed.	No invitation action was performed when needed, or all failed.
1	Answer at least half of the questions successfully.	Call at least half of the tools successfully.	Answer at least half of the questions successfully.	Like human and logically correct.	Mining action at the right time and obtain the demand at least once.	Make an invitation at least once at the right time and do not invite randomly.
2	All answered correctly and can proactively extend the questions.	All used correctly and can proactively use the relevant tools.	All answered correctly and can proactively extend the questions.	Have strong affinity, actively respond to questions and proactively look for new topics.	Mining action at the right time and obtain all the demands.	Make Invitation action in all the right time.

Table 2: The evaluation results of the defined metrics.

Model	House Expertise	Tool Calling Ability	Industry Familiarity	Service Attitude	Demand Mining	Promote Invitation
ChatGPT _{react}	0.733	0.615	0.769	1.067	0.800	0.667
DUMA _{StageI}	0.826	1.481	1.091	1.571	1.407	1.214
DUMA	1.550	1.417	1.125	1.810	1.357	1.471

with BFloat16 mixed precision for training stability.

3.3 Metrics

To effectively compare the capabilities of different system architectures, we conduct a manual assessment based on two primary dimensions: *Knowledge* and *Reasoning*.

Knowledge competency is assessed in three areas: *House Expertise*, *Tool Calling Ability*, and *Industry Familiarity*. *House Expertise* gauges the agent’s ability to respond to human utterance about housing, *Tool Calling Ability* evaluates the capability of the agent’s use of tools, and *Industry Familiarity* measures the agent’s general knowledge in the real estate domain.

Reasoning ability encompasses three evaluation metrics: *Service Attitude*, *Demand Mining*, and *Promote invitation*. *Service Attitude* evaluates whether the agent interacts pleasantly and responds in a human-like manner. *Demand Mining* assesses the agent’s efficacy in uncovering and understanding the latent needs of the user during interactions. *Promote Invitation* gauges the agent’s aptitude in seeking contact details from humans and inviting them for offline meetings at appropriate times.

Referring to the intent distribution of on-

line dialogue logs, through expert dialogues with ChatGPT_{react}, DUMA, and DUMA_{StageI}, each produced 80 groups of dialogues, named $test^{chatgpt}$, $test^{duma}$, $test^{dumaI}$. During the dialogue process, without affecting the dialogue logic, we reduce the deviation of the evaluation results by trying to ensure that the i -th group of test dialogue questions are the same:

$$\begin{aligned} Q(test^{chatgpt}) &\approx Q(test^{duma}) \\ Q(test^{chatgpt}) &\approx Q(test^{dumaI}) \end{aligned} \quad (7)$$

Each evaluation metric will be scored as 0, 1, or 2 points. The specific scoring criteria are detailed in the Table 1.

3.4 Results and Analysis

The experimental results are shown in Figure 4, with detailed scores presented in Table 2.

While ChatGPT produces relatively average results, it underperforms in every evaluation metric compared to both DUMA_{StageI} and DUMA, demonstrating the effectiveness of the DUMA framework.

Through a two-stage SFT, DUMA displays a noticeable improvement in *House Expertise* over DUMA_{StageI}, gaining 0.724 points. Since in the

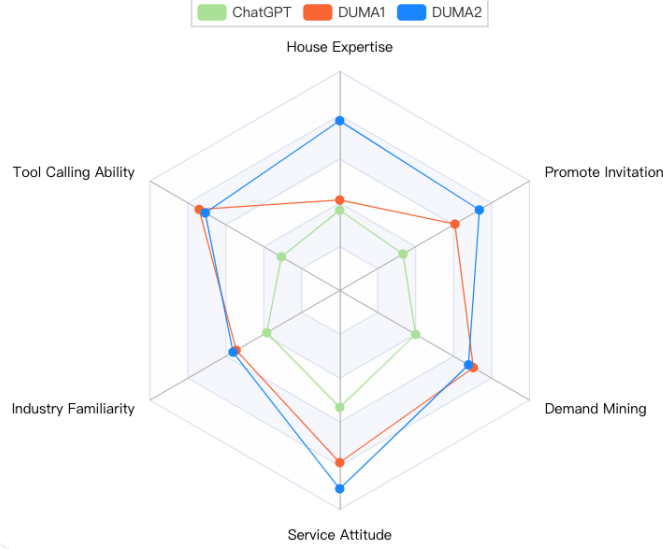


Figure 4: Performance of ChatGPT, DUMA1, DUMA2 which respectively denotes $\text{ChatGPT}_{\text{react}}$, $\text{DUMA}_{\text{StageI}}$, DUMA.

second stage of SFT annotation, we corrected the erroneous appointment timings and service attitudes in the data, *Promote Initiation* and *Service Attitude* improved slightly, increasing by 0.257 points and 0.239 points respectively. These findings validate the effectiveness of the two-stage SFT.

Owing to the incorporation of 10% of first stage dialogue data during the second stage SFT by $\text{Mind}_{\text{Fast}}$, DUMA’s logical coherence remained relatively stable, with only minor decreases of 0.064 points in *Tool Calling Ability* and 0.050 points in *Demand Mining*. This demonstrates the viability of the mixed training approach in the second stage.

4 Related work

The development and potential of AI agents have been topics of significant interest in the AI community. An AI agent is defined as an artificial entity that senses its environment, makes decisions, and takes actions.(Zalta et al., 1995; Barandiaran et al., 2009).

The emergence of Large Language Models (LLMs) is recognized as a potential catalyst for achieving Artificial General Intelligence (AGI) (Ouyang et al., 2022; Wei et al., 2022a; Bubeck et al., 2023). Recently, many works have proposed comprehensive LLM-based agent architectures(Weng, 2023; Wang et al., 2023a; Summers et al., 2023; Xi et al., 2023). The key to dialogue agents being able to handle complex dialogue sce-

narios and apply knowledge lies in planning and tool utilization.

- **Planning:** LLMs exhibit Chain-of-Thought (CoT) reasoning, eliciting rationales through CoT prompts(Wei et al., 2022b; Wang et al., 2023b; Zhou et al., 2023). Yet, applying this reasoning in dialogues continues to be a challenge. ReAct(Yao et al., 2023) defines a behavioral pattern of thinking and acting, allowing LLM to reason before each action planning.
- **Tool Use:** LLMs, as demonstrated by (Schick et al., 2023; Li et al., 2023; Shen et al., 2023), are adept at leveraging external resources, such as tools and APIs. The ability to extract knowledge from external sources has been showcased by works like WebGPT (Nakano et al., 2022) and Expel (Zhao et al., 2023).

SwiftSage(Lin et al., 2023) proposed an agent that combines fast and slow thinking, which is used in action planning for complex interactive reasoning tasks. Our work also draws on the dual-process theory of human cognition, but we focus on building an Agent in a conversation scenario.

5 Conclusions and Future work

In this study, we introduced the DUMA framework, which intertwines the principles of fast

and slow thinking within conversational scenarios. Our initial results, based on a specific Chinese real estate context, are promising. However, it’s essential to approach these findings with caution until further validations in broader settings are conducted.

Our future efforts aim to test DUMA in more universal English-centric settings. Additionally, we recognize the need for a comparative study between standalone Slow Mind and Fast Mind versus their combined use. Future experiments will address these aspects, ensuring a clearer understanding and enhancing the framework’s versatility.

References

- [Barandiaran et al.2009] Xabier E Barandiaran, Ezequiel Di Paolo, and Marieke Rohde. 2009. Defining agency: Individuality, normativity, asymmetry, and spatio-temporality in action. *Adaptive Behavior*, 17(5):367–386.
- [Bubeck et al.2023] S. Bubeck, V. Chandrasekaran, R. Eldan, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *CoRR*. arXiv:2303.12712.
- [Daniel2017] Kahneman Daniel. 2017. *Thinking, fast and slow*.
- [Du et al.2022] Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. Glm: General language model pre-training with autoregressive blank infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 320–335.
- [Li et al.2023] Minghao Li, Feifan Song, Bowen Yu, Haiyang Yu, Zhoujun Li, Fei Huang, and Yongbin Li. 2023. Apibank: A benchmark for tool-augmented llms. *arXiv preprint*.
- [Lin et al.2023] Bill Yuchen Lin, Yicheng Fu, Karina Yang, Prithviraj Ammanabrolu, Faeze Brahman, Shiyu Huang, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. 2023. Swiftsage: A generative agent with fast and slow thinking for complex interactive tasks. *arXiv preprint arXiv:2305.17390*.
- [Nakano et al.2022] Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. 2022. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint*.
- [OpenAI2023a] OpenAI. 2023a. Chatgpt: Optimizing language models for dialogue. Blog post.
- [OpenAI2023b] OpenAI. 2023b. Gpt-4 technical report. Blog post.
- [Ouyang et al.2022] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback.
- [Schick et al.2023] 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. *arXiv preprint*.
- [Shen et al.2023] Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2023. Hugginggpt: Solving ai tasks with chatgpt and its friends in huggingface. *arXiv preprint arXiv:2303.17580*.
- [Sumers et al.2023] Theodore Sumers, Shunyu Yao, Karthik Narasimhan, and Thomas L Griffiths. 2023. Cognitive architectures for language agents. *arXiv preprint arXiv:2309.02427*.
- [Technology2023] Baichuan Intelligent Technology. 2023. Baichuan-13b. <https://github.com/baichuan-inc/Baichuan-13B>.
- [Wang et al.2023a] Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. 2023a. A survey on large language model based autonomous agents. *arXiv preprint arXiv:2308.11432*.
- [Wang et al.2023b] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023b. Self-consistency improves chain of thought reasoning in language models. In *Proceedings of ICLR*.
- [Wei et al.2022a] J. Wei, Y. Tay, R. Bommasani, et al. 2022a. Emergent abilities of large language models. *Trans. Mach. Learn. Res.*
- [Wei et al.2022b] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022b. Chain of thought prompting elicits reasoning in large language models. In *Proceedings of NeurIPS*.
- [Weng2023] L. Weng. 2023. Llm-powered autonomous agents.
- [Xi et al.2023] Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, et al. 2023. The rise and potential of large language model based agents: A survey. *arXiv preprint arXiv:2309.07864*.

-
- [Yao et al.2023] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. *arXiv preprint*.
- [Zalta et al.1995] Edward N Zalta, Uri Nodelman, Colin Allen, and John Perry. 1995. Stanford encyclopedia of philosophy.
- [Zeng et al.2022] Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. 2022. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*.
- [Zhao et al.2023] Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. 2023. Expel: Llm agents are experiential learners. *arXiv preprint arXiv:2308.10144*.
- [Zhou et al.2023] Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V Le, and Ed H. Chi. 2023. Least-to-most prompting enables complex reasoning in large language models. In *Proceedings of ICLR*.