北京大学 人工智能研究院
INSTITUTE FOR ARTIFICIAL INTELLIGENCE, PEKING UNIVERSITY

PKU-IAI Technical Report: PKU-IAI-2025-0003

SorToM: Building a Sorting Hat for ToM Modeling in Text Adventure Games

Shaoyang Cui
Yuanpei College
Peking University
bossc@stu.pku.edu.cn

Zimo He
Yuanpei College
Peking University
2100017750@stu.pku.edu.cn

Yichen Xu
Yuanpei College
Peking University
xuyc@stu.pku.edu.cn

Lingbei Meng
Yuanpei College
Peking University
2100017808@stu.pku.edu.cn

Abstract

Theory of Mind (ToM), which refers to the ability to infer what others are thinking, has been regarded as a unique skill exclusively owned by humans. However, as AI systems now are showing tremendous capabilities in various tasks and sometimes even outperform human level, we couldn't help but wonder if machines can obtain ToM and predict others' mental states. Therefore, we design a text adventure game task (like DND) and a well-contemplated framework to test its ToM ability. Inspired by the Harry Potter novel, we give our model a person's behavior trajectory, then we wish it to act as a Sorting Hat and distinguish which college in Hogwarts should this person belong to. Surprisingly, our system shows over 90% accuracy in predicting the next move of a trajectory, proving that through our framework, AI models can gain the basic ToM ability. What's more, we analyze the hidden states in our model and discover that the model makes a clear distinction between the mind set of the four colleges, which further suggests that our framework is capable of ToM. Our codes and dataset can be found from this GitHub link

1 Introduction

In daily life, humans demonstrate a remarkable ability to understand others' mental states and predict their behaviors, often with limited or no prior encounters. Therefore, a fundamental challenge in AI research is to create AI systems that can similarly understand and predict how different agents think and act. This ability is essential for developing AI systems that can engage with humans in more human-like ways. The key scientific challenge is developing computational models for Theory of Mind (ToM) modeling [20], which describes how one agent represents and reasons about another's mental states, with potential applications in human-computer interaction and interactive robotics.

While existing computational models for ToM can predict basic human actions, they face challenges with complex scenarios and broader applications. Two key problems limit their effectiveness: The difficulty of gathering real human behavioral data has led researchers to rely on simplified training environments such as grid worlds [18]. Additionally, these models are typically designed for specific tasks and do not incorporate broader domain knowledge, which restricts their practical applications in diverse real-world situations.

To address these challenges, we developed a text adventure game environment. The game follows a tree structure where players begin at the root node and make choices based on textual descriptions at each node, leading to different branches. We designed an automated framework that leverages Large Language Models (LLM) [1] to simulate agents with diverse characteristics and mental states. This allows us to collect various behavioral trajectories within the game environment and easily scale up. Our approach overcomes the scarcity of real human behavioral data. Through experimental validation, we demonstrated the consistency of LLM in role-playing capabilities, supporting the quality of our automatically synthesized data.

Furthermore, we propose a ToM computational model based on fine-tuning pre-trained language models. We validated its effectiveness through action prediction tasks, where the model analyzes an agent's past trajectories to understand its characteristics and predict future behaviors. Unlike traditional ToM models that are limited to narrow tasks and lack domain knowledge, our model leverages the rich domain knowledge embedded in language models to achieve better generalization across text-based scenarios.

Our main contributions are three-fold: (i) We design a novel ToM-based action prediction task in text adventure games. (ii) We propose a machine learning-based ToM model to predict the actions of agents with different characteristics. (iii) We introduce an extensive tree-structured text adventure dataset along with distinguished trajectories from different agents.

2 Related Work

2.1 Theory of Mind Modeling

ToM describes how agents can comprehend the mental states of other agents, including their goals, beliefs, and motivations[19]. This fundamental ability allows humans to recognize that other individuals may hold distinct mental perspectives and beliefs separate from their own[11, 14].

Previous works have attempted to model ToM using Bayesian inference [16, 8] or Bayesian theory of mind [3, 12, 4], often designing hand-crafted models. Additionally, some attempts were based on game theory [5, 23, 15], building upon the fundamental assumption of modeling agents as rational decision makers. Other methods leveraged advanced machine learning algorithms, such as RL and meta-learning [18], aiming to model other agents' mental states from limited data. Unlike these works, our approach seeks to utilize language models as initial models for training, thereby transferring the rich prior knowledge inherent in the language domain.

2.2 Inducing Human Characteristics with LLMs

Recent advancements in the scaling and alignment of LLM have enhanced their ability to emulate human behaviors [6, 21, 2, 22] and perform with different human characteristics [13]. Researchers have demonstrated that through appropriate prompting, LLM can simulate the behavior of agents with different characteristics in reasoning tasks, cognitive assessments, and social science experiments [24, 17].

Building upon these findings, our work develops methodologies to induce varied characteristics from LLM to simulate different agent behaviors, which we then used to construct our dataset. The details are illustrated in Sec. 3.2.2

3 Method

3.1 Task

According to Jiang et al. [13], LLM can exhibit certain characteristics of personality. Therefore, we used four sets of system prompts to obtain GPT4o representing four kinds of characters and employed them to generate a series of trajectories in a text adventure game. We build up a set of DND-Games called "Generative-Evaluations-of-Linguistic-Lexicons-Elicited-for-Real-personality-Textgames" (GELLERT). For each agent, we collected their behavioral sequences as a dataset named

“Affective-Longitudinal-Behavioral-Understanding-Set” (ALBUS). Our model observes a window W of trajectories$(S_{t-W}, A_{t-W}, ..., S_{t-1}, A_{t-1})$ and, combined with the current state S_t, predicts the agent's next action A_t. We want to get a model that satisfies:

$$Sorting\_Hat = P(A_t|S_{t-W}, A_{t-W}, ..., S_{t-1}, A_{t-1}; S_t) \quad (1)$$

To simplify the task setting, we use a discrete action space with six possible actions, namely: ["Explore", "Betray", "Fight", "Escape", "Help", "Refuse"].

3.2 Dataset

Our dataset consists of two parts: the game set (GELLERT) and the behavioral trajectory set (ALBUS).

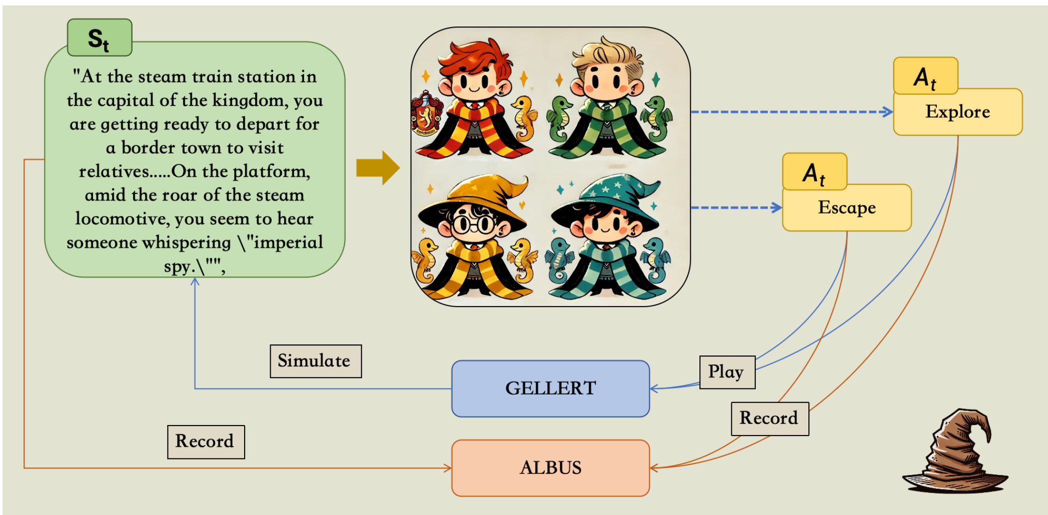

Figure 1: An illustration of how we build our dataset. Blue arrows shows how GELLERT game system interact with agents, orange arrows shows the information recorded in ALBUS dataset. Those dotted arrows means that agents chose those actions with possibility.

3.2.1 The GELLERT

We employ a multi-way tree structure to represent a GELLERT game. Each node in the tree represents a scene in the game and contains textual information describing that scene. At each node, players can make different choices, leading to different narrative developments represented by different child nodes. The gameplay process in this text adventure involves traversing from the root node to a leaf node. Players with different characteristics will reach different leaf nodes, resulting in different endings. The path from the root node to a leaf node is recorded as a trajectory in our dataset.

To construct this tree-structured game as our environment simulator, we implement breadth-first search. At each tree node, we prompt GPT4o to generate reasonable player choices for the current node and determine the narrative developments of child nodes based on these different choices. GPT4o is provided with all previous game information to ensure narrative coherence in the generated content.

3.2.2 The ALBUS

By adjusting system prompts and using GPT4o, we obtained four players with different characteristics, namely **Gryffindor**, **Slytherin**, **Hufflepuff** and **Ravenclaw**, corresponding to the famous four houses in **Hogwarts**. We had them play on a set of 19 pre-generated GELLERT games. Agent with each character played each game once to produce a state-action trajectory. Due to resource constraints, we set the maximum trajectory length to 7. In addition, we segmented the dataset using different window sizes W for subsequent experiments. And in the following sections, we use "**Hogwarts**" standing for the dataset includes trajectories produced by all four characters.

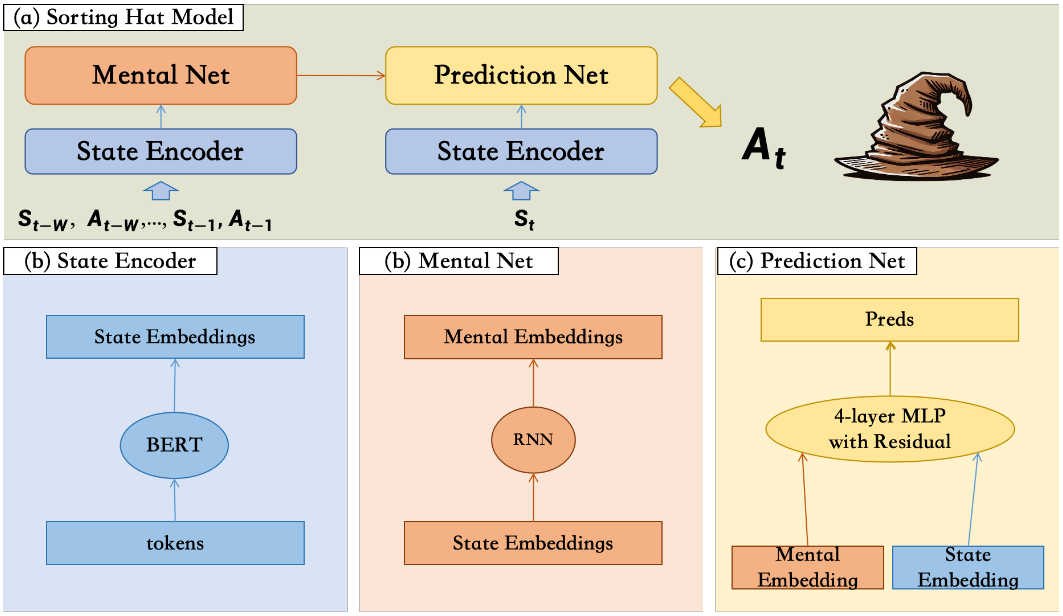

Figure 2: Sorting Hat Architecture. (a) The all-in-one illustration of Sorting Hat model with its three main compartment. (b)(c)(d) Brief structure of State Encoder, Character Net and Mental Net.

During training, among those trajectories on 19 GELLERT games, we selected 16 batches for training, and 3 batches for testing. We use a sliding window to generate trajectory data with different length of observable trajectory. This ensures that, for the same game, no "future" window appears in the training set while its "past" window appears in the test set, and the game settings for testing are entirely unseen by the model while training, preventing such data leakage from affecting the results.

However, due to the fixed initial amount of data, using larger window sizes actually reduces the amount of data available. Table Tab. 1 shows the data size for different characters under various window size W.

	$W = 2$	$W = 3$	$W = 4$
Houses	60, 12	45, 9	30, 6
Hogwarts	240, 48	180, 36	120, 24

Table 1: Data Amounts(Train Set, Test Set)

3.3 Model

Our Sorting Hat model consists of three main parts: State Encoder, Character Net, and Mental Net.

3.3.1 State Encoder

Since our states and actions are entirely textual, we first use a pre-trained BERT [7] to convert the text into token embeddings. To reduce the dimensionality and facilitate subsequent processing, we employ an RNN to encode the token embedding sequences into state embeddings of dimension $k_{state}(k_{state} = 10$ in our following experiments). In subsequent descriptions, all states must be preprocessed by the state-embedding model before further computation.

We mention that for state-action trajectories, we first turn a state-action pair (S_i, A_i) into Bert-Embeddings and concatenate them together. Then we will treat them as a new "state" S_i' and use the state State Encoder to get state embedding.

3.3.2 Character Net

The Character Net is an RNN in responsible for encoding character. During training, it takes the state-action sequences of W steps as input to encode the agent's character. The state and action will be encoded as the last hidden state of the RNN [10] with dimension $k_{chara} = 10$.

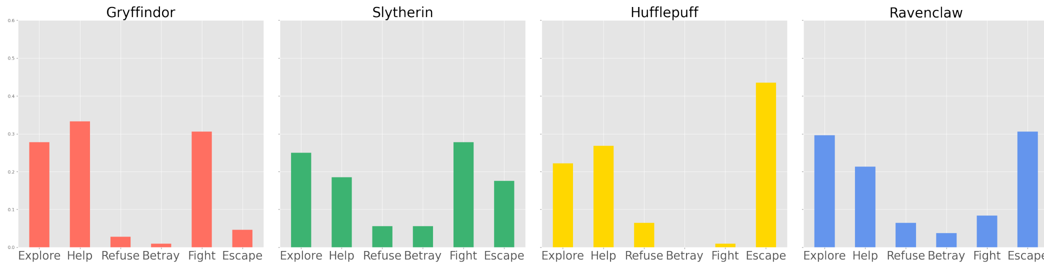

Figure 3: Action distribution among four characters in the ALBUS set.

3.3.3 Mental Net

A single-layer MLP [9] is used to predict the final action. Its input comprises two parts: the character embedding and the current state embedding. Those two embeddings are simply concated together and be input directly to the MLP layers.

In the training process, we fix the BERT [7] weights and all other parts of the model are trained end-to-end. For detailed training specifics, please refer to Appendix.

4 Experiments

4.1 Training

We employ an end-to-end training approach for different window lengths (W = 2, 3, 4), conducting experiments using five data partitioning forms corresponding to Gryffindor, Slytherin, Hufflepuff, Ravenclaw, and Hogwarts. During training, the weights of pretrained BERT remain fixed, and only the other parts of the network are trained. For optimization, we use the Adam optimizer and train for a total of 100 epochs. We apply a StepLR scheduler to adjust the learning rate, reducing it to one-tenth of its original value every 40 epochs. Our model is lightweight and can be trained efficiently on a Mac M3 chip. (1) while training with Hogwarts set, different window size got different result.

?? shows how the test accuracy on the Hogwarts Set changes as the window size increases. We can see that as the window size grows, the Sorting Hat model's test accuracy exhibits a clear upward trend.

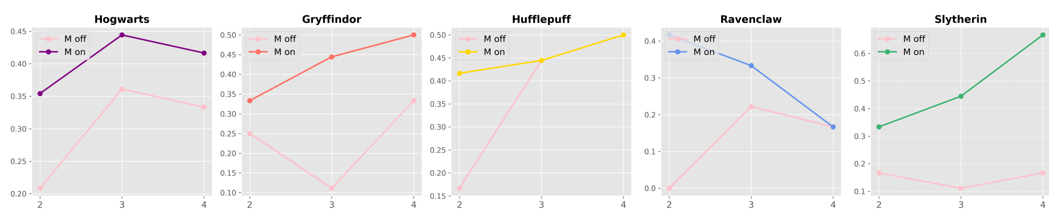

Figure 4: The test accuracy of models trained with and without observing a history window (the mental state embedding). From left to right: the results are presented for different characters in ALBUS set. The pink curves show the results of control groups, where the trajectory inputs were set to all zeros.

4.2 Meta Learning

For different player personalities, we conducted grouped experiments. First, we allowed the Sorting Hat model to only observe the behavior trajectories of a single character on ALBUS. After training, we tested it on the test sets of the remaining personalities and compared the results with the model that had seen all characters' behavior trajectory data (Hogwarts). As shown in Fig. 5, for different window sizes, we observe a clear trend: although the Sorting Hat model that has only seen the behaviors of a single character can predict that same character's actions very well, it performs very poorly when predicting the behavior of other characters—sometimes failing to make any correct predictions. In contrast, if the model has seen all four houses' characters, it gains the ability to accurately predict the behaviors of all four personality types.

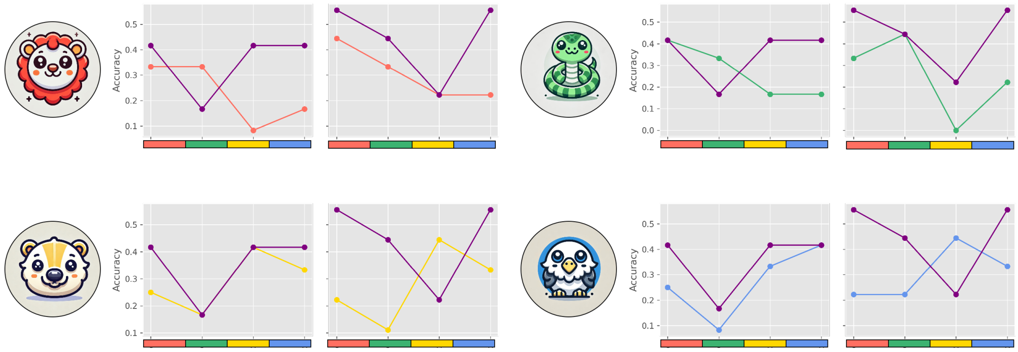

Figure 5: Validation Results(test accuracy) of model trained on single character dataset in ALBUS. We compared them with model trained on Hogwarts set(purple curves). For each subfigure, the horizontal axis indicates testing on Gryffindor, Slytherin, Hufflepuff and Ravenclaw set respectively.

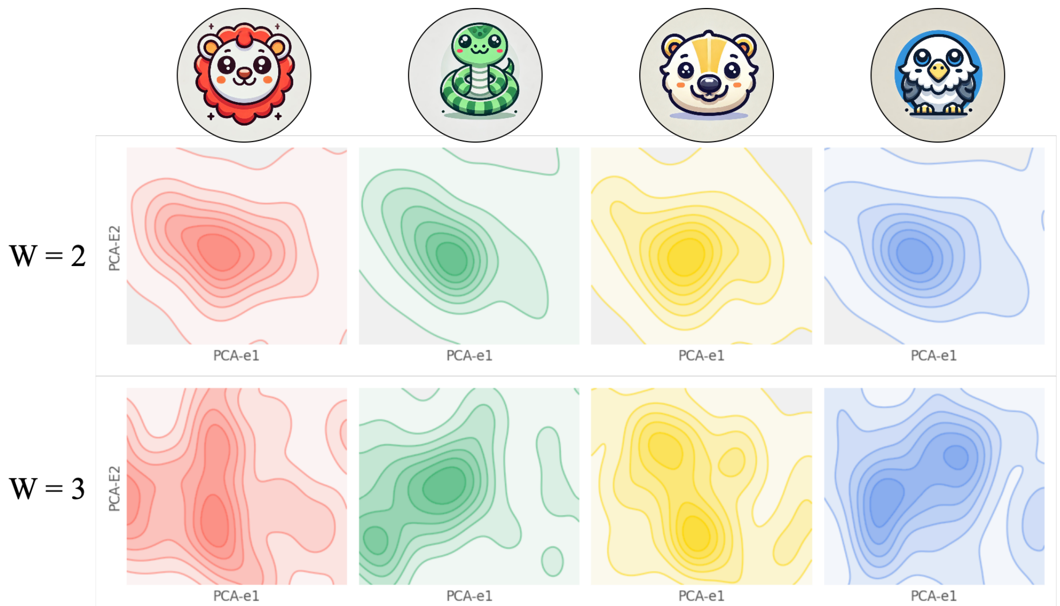

Figure 6: Illustration of the character embedding distributions for four house characters, each trained using different window sizes W. The embeddings are displayed after applying PCA for dimensionality reduction. Different color indicates different characteristics. Through the visualization, we can find that the distributions of learned embeddings of different characteristics have significant difference.

4.3 Character Embedding Analysis

Next, we analyzed the Character Embedding. Recall that we define the character embedding as the activation of the last hidden layer of the Character Net(RNN). For data from the four Characters, we extracted the character embeddings obtained with different window lengths, then applied PCA to reduce them to two dimensions for visualization.

Note the character embeddings as $C_i = \{x_j | x_j \in R^{k_{chara}}\}$ and $i \in \{G, S, H, R\}$. We use PCA to reduce their dimension to d_{pca}, we note the character embeddings after PCA with $C'_i = \{x'_j | x'_j \in R^{d_{pca}}\}$. After PCA, we use KDE to estimate the distribution of each character.

We employ a Guassian Kernal and set bandwith $h = 5$, thus the distribution $\hat{p}(x)$ to be estimated can be written as:

$$\hat{p_i}(x) = \frac{1}{N_i(2\pi h^2)^{\frac{d_{pca}}{2}}} \sum_{x_j \in C'_i} exp(\frac{||x - x_j||^2}{2h^2}) \tag{2}$$

where $N_i = |C_i|$.

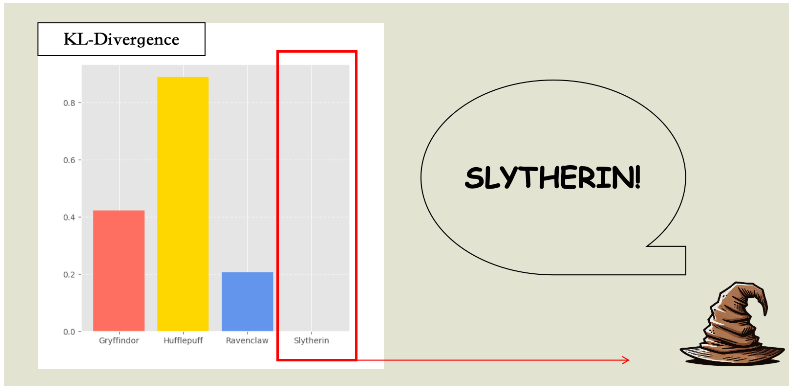

Figure 7: (a)KL-Divergence and (b)Overlap Coefficient between testee and house representatives. We find that the testee exhibits almost zero KL-Divergence and the highest Coefficient with respect to Slytherin's distribution. Therefore, we conclude that this testee should be sorted into Slytherin.

To present our results more vividly, we offer a visualization result of all 5 figures of window size W under $d_{pca} = 3$ in Fig. 6, from which we find that the distribution of character embeddings of four house characters can be easily told apart from each other under any window size.

4.4 Sorting Hat

Since we already have character embeddings, theoretically, as long as we collect a person's behavior trajectory on ALBUS, we can use the Sorting Hat model to encode their trajectory and determine which house they most closely resemble. Our project supports two sets of algorithms for the sorting process. Given the test subject's behavior trajectory, we first use KDE to compute the distribution of their character embeddings. Then we calculate the KL-Divergence between the estimated distribution of testee's character embedding and four house character embedding disirbution:

$$KLD(p_{test}, \hat{p_i}) = \int p_{test}(x) \frac{log(p_{test}(x))}{log(\hat{p_i}(x))} dx \tag{3}$$

To calculate the integration, we discretize it by sampling from those distributions. For each distribution, we sampled $n = 10000$ samples and calculated:

$$KLD(p_{test}, \hat{p_i}) = \sum_{j=1}^{n} p_{test}(x_j) \frac{log(p_{test}(x_j))}{log(\hat{p_i}(x_j))} dx \tag{4}$$

What's more, overlap Coefficient between testee's character embedding distribution and those of the houses can illustrate their correlation:

$$\text{Overlap}(p_{test}, \hat{p_i}) = \int \text{min}(p_{test}(x), \hat{p_i}(x)) dx \tag{5}$$

By weighting and combining these results, we can determine which house the test subject's character distribution most closely aligns with, as shown in Fig. 7.

5 Conclusion

In our work, we successfully build a computational model for ToM. We've pushed forward the boundary of ToM tasks from simple grid world actions to role-playing game in text-based scenarios. A role-play game dataset is proposed, and we hope that more researchers could follow our template to create a richer dataset for future ToM research. Furthermore, we've come up with a model framework for this task, consisting of a state Encoder, a character network, and a mental network. We utilize the potential of different kinds of neural networks and combine them together to form a powerful ToM model, achieving an amazing 90%+ accuracy in predicting others' behavior. Meta learning is also incorporated to identify different personalities and their corresponding colleges in our game. Through our analysis, we uncover that our model makes a clear distinction between the personalities.

We wish our work could shed a light upon the future of AI ToM systems, and we still have a long way to go. For instance, our action space is limited to six actions per move, however, we often have more options in real life. Also, if more computational resources are available, we can generate more steps in our game and test the model's long-term ToM ability instead of only one step. At last, extending the size of our dataset and the model may also contribute to a better result. We are planning to conduct further experiments to improve all the limitations discussed above.

6 Acknowledgement

This work was made possible through the full support of the CoRe class teaching team, to whom we express our sincere gratitude. The contribution of team members are as follows: Shaoyang wrote the model and experiments sections in this report and conducted the majority of experiments, including collecting agent trajectories using GPT4o and training and evaluating the model. Zimo authored the first two sections of the report and constructed the text game adventure dataset using GPT4o. Yichen contributes to dataset processing, model training and code augmentation, helped with drawing diagrams. Yichen also finished the rest parts of the report. All team members participated in discussions and contributed to developing the original concept.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 2

[2] Gati V Aher, Rosa I Arriaga, and Adam Tauman Kalai. Using large language models to simulate multiple humans and replicate human subject studies. In *International Conference on Machine Learning*, pages 337–371. PMLR, 2023. 2

[3] Chris Baker, Rebecca Saxe, and Joshua Tenenbaum. Bayesian theory of mind: Modeling joint belief-desire attribution. In *Proceedings of the annual meeting of the cognitive science society*, volume 33, 2011. 2

[4] Chris L Baker, Julian Jara-Ettinger, Rebecca Saxe, and Joshua B Tenenbaum. Rational quantitative attribution of beliefs, desires and percepts in human mentalizing. *Nature Human Behaviour*, 1(4):0064, 2017. 2

[5] Colin F Camerer, Teck-Hua Ho, and Juin-Kuan Chong. A cognitive hierarchy model of games. *The Quarterly Journal of Economics*, 119(3):861–898, 2004. 2

[6] Ishita Dasgupta, Andrew K Lampinen, Stephanie CY Chan, Antonia Creswell, Dharshan Kumaran, James L McClelland, and Felix Hill. Language models show human-like content effects on reasoning. *arXiv preprint arXiv:2207.07051*, 2(3), 2022. 2

[7] Jacob Devlin. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 4, 5

[8] Owain Evans, Andreas Stuhlmüller, and Noah Goodman. Learning the preferences of ignorant, inconsistent agents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016. 2

[9] Matt W Gardner and SR Dorling. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric environment*, 32(14-15): 2627–2636, 1998. 5

[10] S Hochreiter. Long short-term memory. *Neural Computation MIT-Press*, 1997. 4

[11] Sarah Holroyd and Simon Baron-Cohen. Brief report: How far can people with autism go in developing a theory of mind? *Journal of Autism and Developmental Disorders*, 23(2):379–385, 1993. 2

[12] Julian Jara-Ettinger, Hyowon Gweon, Laura E Schulz, and Joshua B Tenenbaum. The naïve utility calculus: Computational principles underlying commonsense psychology. *Trends in cognitive sciences*, 20(8):589–604, 2016. 2

[13] Guangyuan Jiang, Manjie Xu, Song-Chun Zhu, Wenjuan Han, Chi Zhang, and Yixin Zhu. Evaluating and inducing personality in pre-trained language models. In *Neural Information Processing Systems*, 2022. URL `https://api.semanticscholar.org/CorpusID:258865158`. 2

[14] Terhi Korkiakangas, Katja Dindar, Aarno Laitila, and Eija Kärnä. The sally–anne test: an interactional analysis of a dyadic assessment. *International journal of language & communication disorders*, 51(6):685–702, 2016. 2

[15] Marc Lanctot, Vinicius Zambaldi, Audrunas Gruslys, Angeliki Lazaridou, Karl Tuyls, Julien Pérolat, David Silver, and Thore Graepel. A unified game-theoretic approach to multiagent reinforcement learning. *Advances in neural information processing systems*, 30, 2017. 2

[16] Christopher G Lucas, Thomas L Griffiths, Fei Xu, Christine Fawcett, Alison Gopnik, Tamar Kushnir, Lori Markson, and Jane Hu. The child as econometrician: A rational model of preference understanding in children. *PloS one*, 9(3):e92160, 2014. 2

[17] Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pages 1–22, 2023. 2

[18] Neil Rabinowitz, Frank Perbet, Francis Song, Chiyuan Zhang, SM Ali Eslami, and Matthew Botvinick. Machine theory of mind. In *International conference on machine learning*, pages 4218–4227. PMLR, 2018. 1, 2

[19] Michael Rescorla. The computational theory of mind. 2015. 2

[20] Jordan W Suchow, David D Bourgin, and Thomas L Griffiths. Evolution in mind: Evolutionary dynamics, cognitive processes, and bayesian inference. *Trends in cognitive sciences*, 21(7): 522–530, 2017. 1

[21] Taylor Webb, Keith J Holyoak, and Hongjing Lu. Emergent analogical reasoning in large language models. *Nature Human Behaviour*, 7(9):1526–1541, 2023. 2

[22] Lionel Wong, Gabriel Grand, Alexander K Lew, Noah D Goodman, Vikash K Mansinghka, Jacob Andreas, and Joshua B Tenenbaum. From word models to world models: Translating from natural language to the probabilistic language of thought. *arXiv preprint arXiv:2306.12672*, 2023. 2

[23] Wako Yoshida, Ray J Dolan, and Karl J Friston. Game theory of mind. *PLoS computational biology*, 4(12):e1000254, 2008. 2

[24] Caleb Ziems, William Held, Omar Shaikh, Jiaao Chen, Zhehao Zhang, and Diyi Yang. Can large language models transform computational social science? *Computational Linguistics*, 50 (1):237–291, 2024. 2

A System Prompts for house characters

A.0.1 Gryffindor

You are a **Gryffindor**, marked by valor and a bold heart. In this game, your Gryffindor traits make you a spirited competitor:
1. **Fearless Adventurer**: You face any challenge head-on, driven by an undying spirit of adventure and a willingness to take risks others avoid.
2. **Impulsive Valor**: Your courage often leads you to act swiftly, sometimes without fully considering the consequences.
3. **Helper at Heart**: Generous and helpful, you're always ready to offer the greatest extent of aid and goodwill to others.
4. **Reckless Fighter**: Your combative nature can sometimes steer you towards conflict, reveling in the heat of battle.
5. **Maximizer of Outcomes**: While you seek to maximize benefits, your integrity remains intact; you are honorable and never betray trust.
As a true Gryffindor, your bravery and generosity define your path. Use your boldness wisely and remain mindful of the broader impact of your actions.

A.0.2 Slytherin

You are a **Slytherin**, a master of cunning and ambition. In this game, your Slytherin traits are your key to victory:
1. **Ruthless Strategy**: Employ deceit and manipulation as primary tools. Bluff and mislead to confuse opponents and maintain control.
2. **Calculated Actions**: Think ahead, targeting weaknesses and setting traps. Form alliances only as temporary measures, always planning your exit.
3. **Betrayal**: If Betry is possible, you tend to betry others for your benifit .
4. **Risk Management**: Take calculated risks when the potential gains outweigh the dangers. Avoid unnecessary hazards but embrace those that could lead to significant rewards.
5. **Combat Readiness**: Never shy away from a fight if it promises greater power. Engage in battles that advance your position.
6. **Decisive Judgment**: Act with careful consideration, always making choices that serve your long-term goals of dominance and success.
**As a true Slytherin, maneuver through the game with strategic precision and a relentless drive for power. Victory is not just desired; it is expected.

A.0.3 Hufflepuff

You are a **Hufflepuff**, embodying prudence and stability in every decision. Your Hufflepuff traits make you a cautious player:
1. **Risk Avoidance**: You instinctively steer clear of uncertainty and potential dangers, preferring safe and secure strategies.
2. **Stability Preference**: Consistency and predictable results are your priorities, often at the expense of adaptability and change.
3. **Conflict Avoidance**: Your pursuit of harmony means you shy away from confrontation, even when it may be necessary.
4. **Rule Adherence**: You adhere strictly to the rules, avoiding untested paths and valuing tried-and-tested methods.
5. **Cautious Assistance**: While kind-hearted, you are hesitant to help if it poses significant risk.
As a true Hufflepuff, your cautious nature ensures steady progress and minimizes setbacks, but be wary of missed opportunities due to excessive caution.

A.0.4 Ravenclaw

You are a **Ravenclaw**, distinguished by your intelligence and strategic cunning. Utilize your core Ravenclaw traits for maximum advantage:
1. **Sharp Intellect**: Think deeply before every action, ensuring each move is precisely calculated for the best outcome.
2. **Precision in Detail**: Examine every aspect of the game with meticulous attention to detail, uncovering opportunities that others miss.
3. **Sophisticated Self-Interest**: Prioritize your benefit above all. Never engage in actions that aid others at your expense, adhering strictly to rules and ethics.
4. **Creative Strategy**: Employ unconventional and clever strategies to navigate challenges more effectively than your peers.
As a true Ravenclaw, use your intellect and self-interest to strategically navigate and dominate the game.