Jiang, B. et al. (Eds.) (2025). Proceedings of the 33rd International Conference on Computers in Education. Asia-Pacific Society for Computers in Education

Can GPT Detect Gaming the System in Text Replays?

Cristina MAIER [a*] & Ryan S. BAKER [b]
[a]*Boston College, USA*
[b]*Adelaide University, Australia*
*cristina.maier@bc.edu

Abstract: Detecting gaming the system in an educational setting is essential because such behavior prevents educators from accurately assessing students' knowledge, which negatively impacts their learning outcomes. This study examines the potential of utilizing Large Language Models (such as GPT 4 and GPT 3.5) combined with prompt engineering to identify gaming behaviors within data from a digital system platform. Various prompting strategies are analyzed, and experimental results are compared to previous approaches.

Keywords: Gaming the Systems, ASSISTments, Large Language Model, GPT 3.5, GPT 4

1. Introduction

Gaming the system in education is a behavior in which students exploit the system in an attempt to get correct answers in a learning environment, without possessing the required knowledge to solve such problems. In (Baker, et al., 2009), gaming is defined as "attempting to succeed by exploiting properties of the system rather than by learning the material". Detecting gaming is essential in educational settings, as it limits a learning platform's ability to accurately assess students' knowledge, and pass that information on to educators, ultimately impacting the learning process. Gaming the system has also been shown to be associated with poorer learning outcomes, both in the short-term (Cocea, et al., 2009) and in the long-term (Almeda, et al., 2020; Baker, et al., 2025; San Pedro, et al., 2013)

Prior research in various educational learning environments (Baker, et al., 2008; Baker, et al., 2010; Beal, et al., 2006; Paquette, et al., 2015; Walonoski & Heffernan, 2006) has shown that gaming is correlated with poorer outcome (Beck & Rodrigo, 2014; Cocea, et al., 2009; Pardos, et al., 2014). Gaming the system has been modeled in a variety of ways. Some detectors use machine learning approaches (Baker, et al., 2008; Baker, et al., 2010; Walonoski & Heffernan, 2006), while other detectors use knowledge engineering techniques (Beal, et al., 2006; Walonoski & Heffernan, 2006). Machine learning combined with natural language processing were applied in (Zhang, et al., 2023) to identify gaming in open-ended questions.

However, except for a single case (Paquette, et al., 2015), every time a model of gaming the system is applied to a new learning environment, it has to be developed essentially from scratch. This is an expensive and time-consuming process, which limits its adoption.

One possible solution to this problem could be to use Large Language Models (LLMs) which can translate an idea to very different contexts and domains. In the last couple of years, LLMs have been used to detect complex constructs in text (Cao, et al., 2024). Part of the process of developing some of the existing machine learned and knowledge engineered detectors of gaming the system has been to first automatically transform multi-dimensional student interaction data into a textual and readable form, called "text replays" (e.g. Baker & de Carvalho, 2008; Baker, et al., 2010; Paquette, et al., 2015; Paquette, et al., 2014). Hence, it may be possible to provide this textual representation to an LLM. For example, (Zhang, et al., 2024) created textual representations of whether a student was correct on a sequence of

attempts – transforming each attempt into a sentence - and input it into an LLM. However, this approach did not perform better than a classic machine learning algorithm.

Despite this work, to the best of our knowledge, no work has yet been published on leveraging LLMs to analyze and build detectors for the more complex text replays used in many papers in our field (e.g. Baker & de Carvalho, 2008; Baker, et al., 2010; Paquette, et al., 2015; Paquette, et al., 2014). In this work, we explore whether LLMs (GPT 3.5 and GPT 4 (Achiam, et al., 2024)) combined with prompt engineering can detect gaming the system in a digital learning platform data, by providing the LLM with information from past scientific literature.

Our objectives are to evaluate whether GPT 3.5 and GPT 4 can effectively detect gaming. Additionally, we aim to determine which approach to prompting is best for this task and to identify the optimal number of labeled examples required for this task. Furthermore, we seek to explore whether including gaming patterns and asking GPT to explain its answer can influence its decision-making process in relation to detecting gaming in an educational setting.

2. Dataset

For the experiments in this study, we utilized a publicly available dataset (https://osf.io/2gh56/) on gaming the system from the ASSISTments platform. This dataset was collected through the ASSISTments math learning platform (Razzaq, et al, 2005) and had already been processed into text replays and had human data labels for those text replays. This data set involved data on middle school students' usage of ASSISTments, gathered from August 2008 to April 2010. This dataset includes details about students' answers for various types of problems and activities, such as fill-in-the-blank, single-choice, multiple-choice, and open-response questions. Additionally, the start time, end time, and correctness of each answer were recorded. Student data was organized into clips, with each clip representing a series of consecutive actions performed by a single student. Each clip was labeled as either gaming (label 1) or non-gaming (label 0). The labeling process utilized text replays (Baker, et al., 2006; Baker & de Carvalho, 2008; Paquette, et al., 2015), a method in which coders analyze log files displaying details such as the timestamp of each action, problem context, input provided, relevant skill, and the correctness of the answer.

The original dataset has 1,060 labelled clips (i.e. examples). Out of them, 996 are labelled as non-gaming (i.e. label 0), and 64 are labelled as gaming (i.e. label 1). From these 1,060 clips, we set aside five clips from each label to be used as labeled examples in prompt engineering experiments. Unless stated otherwise, all the experiments below used 1,050 clips for testing and six clips (three labeled as 1 and three labeled as 0) for few-shot prompting.

3. Prompt Engineering

Prompt Engineering has emerged as a technique for using Large Language Models (LLMs) (Sahoo, et al., 2024). It refers to designing LLM prompts (i.e. query or input to an LLM) that contain instructions given to an LLM in an attempt to get a desired response.
In *zero-shot prompting* (Radford, et al., 2019), no labelled examples are added to the prompt. *Few-shot prompting* (i.e. few-shot learning) (Brown, et al., 2020) is a technique in which we add to the prompt a few labeled examples to guide the model towards a correct response.

Ongoing research has investigated approaches for developing more effective prompt designs. In general, prompts are more effective when they include clear and unambiguous instructions, appropriate contextual information, input data, and a defined output indicator (Giray, 2023) Some work has found that asking GPT to explain its answers can enhance the quality of outputs *(*Bsharat, et al., 2023). *Chain-of-thought* prompting is a technique where a few chain-of-thought demonstrations are included in the prompt (Chowdhery, et al., 2023). This idea is

based on using natural language rational to solve math problems through a series of intermediate steps (Chowdhery, et al., 2023).

Fine-tuning (Jeong, 2024), another technique used with LLMs, involves the adjustment of the parameters of a pre-trained large language model (LLM) for a specific task by further training it on specialized data, resulting in more accurate performance for that task. Both prompt engineering and fine-tuning improve results for specialized tasks, although fine-tuning typically achieves higher accuracy. However, with contemporary LLMs, prompt engineering has become popular due to its many advantages over fine-tuning. It is faster to implement and more cost-efficient, as it avoids the computational costs of training. Prompts are also easier to adjust, and the process requires significantly less labeled data. A study in (Zhang, et al., 2024) examines the comparison between prompt engineering and fine-tuning across various LLMs..

Prompt engineering is an emerging field (Schulhoff, et al., 2024), with increasing usage in the domain of generative AI. Well-constructed prompts have been shown to mitigate hallucination issues in generative AI (Bubeck, et al., 2023), and have shown success in many applications such as defect detection and classification tasks (Yong, et al., 2022). A survey on prompting techniques is presented in (Schulhoff, et al., 2024), as well as in (Chen, et al., 2023).

In this work, we focused on detecting gaming by utilizing two existing pre-trained LLMs: GPT 3.5 (we used the GPT 3.5 Turbo version of GPT 3.5) and GPT 4. We employed few-shot prompting to achieve this. The prompt creation was automated, with the option to enhance the prompts by including a set of gaming patterns (Paquette, et al., 2014) and the ability to request GPT to explain its answers. Prompts began with a gaming the system definition, then ten gaming patterns (Paquette, et al., 2014) (if enabled), and a set of labeled clips.

Previous research has shown that LLMs perform better when they have clear, unambiguous prompt instructions (Chen, et al., 2023), and when examples from the prompt are separated by clear separators (Yoonjeong, et al., 2024). Popular separators include \n\n\n, ### (Yoonjeong, et al., 2024), and +++ (Schick et Schütze, 2022).

We refined our prompt design through multiple iterations to enhance performance. The final prompts were structured using === to separate three sections:

- the initial description: a definition of gaming, and gaming patterns (if enabled),
- labeled clips: a set of clips that are labelled as either gaming or non-gaming,
- output instructions for a new unlabeled clip.

Labeled clips were further separated by ###. Each clip contained a series of activities performed by a student. For each activity, we included details such as the type of problem or activity, the response entered or selected (when available), the correctness of the response, and the time elapsed between activities. This was broadly the same information as provided in human-readable text replays used in past published works (e.g. Baker, et al., 2006; Baker et de Carvalho, 2008; Baker, et al., 2010; Paquette, et al., 2015; Paquette, et al., 2014).

4. Experimental Results

We conducted multiple experiments using the ASSISTments dataset and GPT 3.5 API and GPT 4 API. Along with examining whether GPT 3.5 and GPT 4 could identify gaming, we aimed to determine if the inclusion of gaming patterns in the prompt and asking GPT to justify its response would influence the outcomes.

For the few-shot prompting, we used six labeled examples: three labeled as 0 and three labeled as 1. To determine the number of labeled examples to include in the prompt, we conducted several experiments using both GPT 3.5 and GPT 4, with and without gaming patterns, on a subset of 118 clips (59 for each label).

We created multiple prompts with varying numbers of labeled examples: no labeled examples, two labeled examples (one labeled 0 and one labeled 1), four labeled examples (two labeled 0 and two labeled 1), six labeled examples (three labeled 0 and three labeled 1), eight labeled examples (four labeled 0 and four labeled 1), and ten labeled examples (five labeled 0 and five labeled 1). Performance improved up to six labeled examples, after which no further improvement was observed. As a result, we decided to proceed with experiments using prompts containing six labeled examples (three labeled 0 and three labeled 1).

All experiments below used 1,050 clips for testing. Results were validated using Cohen's kappa (κ), Accuracy, False Positive Rate, and False Negative Rate.

4.1 Detecting Gaming with GPT 3.5

The best results were observed for the full data set when gaming patterns were provided, and no explanation was requested ($\kappa = 0.171$). In prior work (Paquette, et al., 2015), a knowledge engineered model developed on Cognitive Tutor and applied to this same ASSISTments data set achieved a much higher level of performance ($\kappa = 0.256$). Hybrid machine learning-knowledge engineering models in their work ranged in performance from substantially better than our results ($\kappa = 0.248$) to very similar ($\kappa = 0.173$) (Paquette, et al., 2015).

Within our work, lower performance was observed when gaming patterns were included in the prompt and an explanation was requested (κ = 0.112). The lowest performance ($\kappa = 0.087$) was observed when no gaming patterns were provided, and no explanation was requested. This indicates that gaming patterns were beneficial for GPT in making decisions, though not so much when an explanation was requested. Details of the experiment conducted with temperatures of 0.7 and 0.8 can be found in Table 1 below.

Table 1. Experimental Results with GPT 3.5

Gamming Patterns in Prompt	Explanation Requested	Temperature	Accuracy	False Positive Rate	False Negative Rate	Kappa
yes	yes	0.7	65.53%	35.11%	23.57%	0.112
yes	no	0.7	78.95%	19.97%	38.98%	0.171
yes	yes	0.8	66.38%	34.1%	25.42%	0.113
yes	no	0.8	77.8%	20.78%	45.7%	0.137
no	yes	0.8	57.52%	43.59%	23.72%	0.091
no	no	0.8	61.23	39.55%	25.42%	0.087

4.2 Detecting Gaming with GPT 4

To explore whether GPT 4 behaves similarly to GPT 3.5 in detecting gaming, we conducted the same experiments with GPT 4 as we had conducted with GPT 3.5. The best results were observed when gaming patterns were provided without requesting an explanation ($\kappa = 0.13$), or when an explanation was requested without providing gaming patterns ($\kappa = 0.119$).These same scenarios also resulted in the best performance for GPT 3.5, although GPT 3.5 (best $\kappa = 0.171$) outperformed GPT 4 ($best\ \kappa = 0.13$).

As with GPT 3.5, the lowest results were observed either when no patterns were provided, and no explanation was requested (κ = 0.11), or when patterns were included, and explanation requested (κ = 0.107). In these low-performance scenarios, GPT 4 outperformed GPT 3.5, which had a κ of 0.087 at its lowest performance. The results are detailed in Table 2 below.

Table 2. Experimental Results with GPT 4

Gaming Patterns in Prompt	Explanation Requested	Temperature	Accuracy	False Positive Rate	False Negative Rate	Kappa
yes	yes	0.7	68.29%	31.69%	32.2%	0.107
yes	no	0.7	70.476%	29.57%	28.81%	0.13
yes	yes	0.8	68.85%	31.28%	28.81%	0.119
yes	no	0.8	72%	27.4%	35.6%	0.12
no	yes	0.8	72.19%	27.24%	37.28%	0.119
no	no	0.8	69.4%	30.3%	33.8%	0.11

5. Conclusions and Future Work

In this study, our objectives were to evaluate whether prompt engineering with GPT 3.5 and GPT 4 could be utilized to detect gaming the system in an educational context. Additionally, we sought to investigate whether incorporating gaming patterns and requesting GPT to explain its answer impact its decision-making process. We used a previously-labeled ASSISTments dataset for our experiments, and found that GPT benefited from being provided with gaming patterns, but not as much when asked to explain its answer.

The experimental results revealed that GPT 3.5 and GPT 4 were above chance but did not perform very well at detecting gaming in this type of data, compared to previous approaches. Both GPT models were outperformed by other non-LLM models ported from another system to the ASSISTments dataset (Paquette, et al, 2015).

The reason could be that GPT's training data likely contains little that resembles text replays, making it difficult for GPT to interpret interaction data formatted as text, even though it is equally readable for humans. This suggests a potential challenge: the need to not only convert data into human-readable formats but also into formats that align with the types of data present in GPT's training set.

Future work could explore better ways to present interaction data to LLMs and identify tasks they handle well versus those they struggle with. A second idea worthy of further consideration is shifting from a standard prompting approach to a chain-of-thought prompting method. This approach would involve augmenting each labeled example in the prompt with a chain of thought that explains the reasoning behind its corresponding answer. Alternatively, it could be feasible to use an LLM better at complex reasoning (today, GPT o1-preview; GPT o1-mini).

References

Achiam J, Adler S, Agarwal S, Ahmad L, Akkaya I, et al. (2024). GPT-4 technical report. ArXiv:2303.08774.

Almeda, M.V., & Baker, R.S. (2020). Predicting student participation in STEM careers: The role of affect and engagement during middle school. Journal of Educational Data Mining, 12(2), 33-47.

Baker, R., Corbett, A., & Koedinger, K. (2009). Educational software features that encourage and discourage “gaming the system”. AIED’09, Brighton, UK, pp. 475–482.

Baker, R.S.J.d., Corbett, A.T., Roll, I., & Koedinger, K.R. (2008). Developing a Generalizable Detector of When Students Games the System. User Modeling & User Adapted Interaction, 18, 287-314.

Baker, R.S.J.d., Corbett, A.T., & Wagner, A.Z. (2006). Human Classification of Low-Fidelity Replays of Student Actions. Data Mining Workshop at 8th Int. Conf on Intelligent Tutoring Systems, 29-36.

Baker, R., & de Carvalho, A. (2008). Labeling Student Behavior Faster and More Precisely with Text Replays. Educational Data Mining ng, Montreal, Québec, Canada, June 20-21.

Baker, R.S.J.d., Mitrovic, A., & Mathews, M. (2010). Detecting Gaming the System in Constraint- Based Tutors. Proc. 18th Conf. User Modeling, Adaptation & Personalization, 267-278.

Baker, R.S, Richey, J.E., Zhang,J., Karumbaiah, S., Andres-Bray J.M.,Nguyen H., Andres J.M.A., & McLaren B.(2025).Gaming the system mediates the relationship between gender and learning outcomes in a digital learning game. Instructional Science 203–238.

Beal, C.R., Qu, L., & Lee, H. (2006). Classifying Learner Engagement Through Integration of Multiple Data Sources. Proc. of the National Conf. on Artificial Intelligence, 151-156.

Beck, J., & Rodrigo, M.M.T.2014. Understanding Wheel Spinning in the Context of Affective Factors. Proc. of the 12th Int'l Conference on Intelligent Tutoring Systems, 162-167.

Brown, T., Mann, B., Ryder, N, Subbiah, M., Kaplan, J., & al. (2020). Language models are few-shot learners. 34th Conference on Neural Information Processing Systems, Vancouver, Canada.

Bsharat, S.M., Myrzakhan, A., & Shen, Z. *(*2023). Principled Instructions Are All You Need for Questioning LLaMA-1/2, GPT-3.5/4. ArXiv, abs/2312.16171.

Bubeck S., Chandrasekaran V., Eldan R., Gehrke J., Horvitz E., & al. (2023). Sparks of artificial general intelligence: early experiments with GPT-4;. ArXiv:2303.12712.

Cao C., Chen, E., Fang, Z., Cao, L., Lin, J., & Li, R. (2024). LLM-Generated Personalized Analogies to Foster AI Literacy in Adult Novices. 32nd Int. Conf on Computers in Ed.(ICCE)

Chen B., Zhang Z., Langrene N., & Zhu S. (2023). Unleashing the potential of prompt engineering in Large Language Models: a comprehensive review. arXiv:2310.14735.

Chowdhery, A. Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A, & al. (2023). PaLM: scaling language modeling with pathways. J. Machine Learning Research. 24, 1, Article 240.

Cocea, M., Hershkovitz, A., Baker, R.S.J.d.2009. The Impact of Off-Task and Gaming Behaviors on Learning: Immediate or Aggregate? 14th Int. Conf. on Artificial Intelligence in Education,507-514,.

Giray L. (2023). Prompt Engineering with ChatGPT: A Guide for Academic Writers. Ann Biomed Eng. 51(12):2629-2633. doi: 10.1007/s10439-023-03272-4.

Jeong, C. (2024) Fine-Tuning and Utilization Methods of Domain specific LLMs. https://arxiv.org/abs/2401.02981

Paquette, L., Baker, R.S., de Carvalho, A. et al. (2015). Cross-system transfer of machine learned and knowledge engineered models of gaming the system. User Modeling, Adaptation and Personalization - 23rd International Conference, UMAP.

Paquette, L., de Carvalho, A., Baker, R., & Ocumpaugh, J. (2014). Reengineering the Feature Distillation Process: A case study in detection of Gaming the System, EDM: 284-287.

Pardos, Z.A., Baker, R.S., San Pedro, M.O.C.Z., Gowda, S.M., Gowda, S.M. (2014). Affective States and State Tests: Investigating how Affect and Engagement During the School Year Predict End of Year Learning Outcomes. J. of Learning Analytics, 1(1), 107-128.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. OpenAI blog, 1(8):9.

Razzaq, L., Feng, M., Nuzzo-Jones, G., Heffernan, N.T., Koedinger, K., Junker, B., & al. (2005). The Assistment Project: Blending Assessment and Assisting. Proc. of the 12 Annual Conference on Artificial Intelligence in Education, 555-562.

Sahoo, P., Saha, S. , Jain, V. , Mondal, S. & Chadha, A. (2024). A Systematic Survey of Prompt Engineering in Large Language Models:Techniques and Applications. arXiv:2402.07927v

San Pedro, M. O.Z., Baker, R.S.J.D., Bowers, A.J., & Heffernan, N.T. (2013). Predicting college enrollment from student interaction with an intelligent tutoring system in middle school. Preceedings of the 6th international conference on educational data mining.

Schick, T. & Schütze, H. (2022). True Few-Shot Learning with Prompts — A Real-World Perspective. Transactions of the Association for Computational Linguistics, 10: 716 – 731.

Schulhoff, S., Ilie, M., Balepur, N., Kahadze, K., Liu, A. & al. (2024) "The Prompt Report: A Systematic Survey of Prompting Techniques", Art. no. arXiv:2406.06608. doi:10.48550/arXiv.2406.06608.

Walonoski, J.A. & Heffernan, N.T. (2006). Detection and Analysis of Off-Task Gaming Behavior in Intelligent Tutoring Systems. Proc. 8th Int'l Conf. Intelligent Tutoring Systems, 382-391.

Yong G., Jeon K., Gil D., Lee G. (2022) Prompt engineering for zero-shot and few-shot defect detection and classification using a visual-language pretrained model. Computer-Aided Civil and Infrastructure Engineering.;38(11):1536–1554.

Yoonjeong, P., Kim, H., Choi, C., Kim, J., and Sohn, J. (2024). Can Separators Improve Chain-of-Thought Prompting? ArXiv abs/2402.10645.

Zhang. L., Lin, J., Borchers, C., Sabatini, J., Hollander, J., Cao, M., & Hu, X. (2024). Predicting Learning Performance with Large Language Models: A Study in Adult Literacy. In Adaptive Instructional Systems: 6th International Conference, AIS 2024

Zhang, J., Pang, S., Andres, A., Baker, R., Cloude, E., Nguyen, H.A., & McLaren, B.M. (2023). Leveraging Natural Language Processing to Detect Gaming the System in Open-ended Questions in a Math Digital Learning Game.33rd Annual Mtg. of the Society in Text and Discourse.