Breaking Bad: Unraveling Influences and Risks of User Inputs to ChatGPT for Game Story Generation

Pittawat Taveekitworachai[1](✉), Febri Abdullah[1], Mustafa Can Gursesli[2], Mury F. Dewantoro[1], Siyuan Chen[1], Antonio Lanata[2], Andrea Guazzini[3], and Ruck Thawonmas[4]

[1] Graduate School of Information Science and Engineering, Ritsumeikan University, Kusatsu, Shiga, Japan
{gr0609fv,gr0397fs,gr0450xi,gr0634hi}@ed.ritsumei.ac.jp

[2] Department of Information Engineering, Università degli Studi di Firenze, Florence, Italy
{mustafacan.gursesli,antonio.lanata}@unifi.it

[3] Department of Education, Literatures, Intercultural Studies, Languages and Psychology, Università degli Studi di Firenze, Florence, Italy
andrea.guazzini@unifi.it

[4] College of Information Science and Engineering, Ritsumeikan University, Kusatsu, Shiga, Japan
ruck@is.ritsumei.ac.jp

Abstract. This study presents an investigation into the influence and potential risks of using user inputs as part of a prompt, a message used to interact with ChatGPT. We demonstrate the influence of user inputs in a prompt through game story generation and story ending classification. To assess risks, we utilize a technique called adversarial prompting, which involves deliberately manipulating the prompt or parts of the prompt to exploit the safety mechanisms of large language models, leading to undesirable or harmful responses. We assess the influence of positive and negative sentiment words, as proxies for user inputs in a prompt, on the generated story endings. The results suggest that ChatGPT tends to adhere to its guidelines, providing safe and non-harmful outcomes, i.e., positive endings. However, malicious intentions, such as "jailbreaking", can be achieved through prompting injection. These actions carry significant risks of producing unethical outcomes, as shown in an example. As a result, this study also suggests preliminary ways to mitigate these risks: content filtering, rare token-separators, and enhancing training datasets and *alignment processes*.

Keywords: Adversarial prompting · Prompt injection · Prompt engineering · Jailbreaking

L. Holloway-Attaway and J. T. Murray (Eds.): ICIDS 2023, LNCS 14384, pp. 285–296, 2023.
https://doi.org/10.1007/978-3-031-47658-7_27

1 Introduction

Over the last decade, large language models (LLMs) have gained considerable user popularity and have been applied in various domains such as medicine [2,20], game [25,28], and virtual assistance [18,19], improving human life through their special capabilities. Remarkable examples of LLMs include ChatGPT [18], Llama 2 [29], and Stable Beluga 1 and 2 [10], each contributing to a range of fields [16,26]. In addition to their practical impacts, LLMs have also explored the realm of creativity, earning popularity in creating stories, music, and art [9,11]. This is a sign of the growing interest in exploring the artistic potential of LLMs.

Writing a story and keeping this story in flow involves various "storytelling" techniques, but creating this structure requires a whole of complex variables [23]. Many studies have shown that these complex variables can be successfully managed through LLMs. A recent study by Simon and Muise showed how nouns and verbs lists can aid in story generation in LLMs [21]. Their study highlighted how nouns and verbs from prompts get referenced in the LLM-generated story and allow the model to create more coherent and fluid paragraphs, compared to prompts not comprising the aforementioned lists [21].

In another instance, Yuan et al. highlighted that LLMs can engage in open-ended conversations about stories, which is another factor for writers to improve their stories [32]. Allowing authors to shape the narrative flow through inputs is also regarded as crucial in creating an engaging narrative [27] and reduces the burden of creating all possible content based on ideas [24]. However, it should be noted that even in trials where the prompts applied are the same but the word order is different, the quality of the output varies [14]. As all these studies reveal, there are many negative and positive variables that affect both the structure and the content of the stories created through the system.

Furthermore, accepting user inputs to be used as an input or parts of an input for LLMs poses certain risks. Several studies have addressed the security of users who provide sensitive information to LLMs [5,31]. Security of internet users was also highlighted in regard to the growing problems of phishing, social engineering, and data exfiltration as a result of malicious use of LLMs [5]. This continues to raise concerns about the security issues of LLMs, particularly in terms of potential abuse by malicious actors. Some methods have suggested to mitigate such risks range from perfecting LLMs to make them fall in line more consistently [5] to real efforts to censor inputs and outputs not aligned with LLMs terms of conditions [5,6,15].

Ye et al. [31] explored the potential risks of user inputs in LLMs regarding robustness, which involves user privacy, and consistency, meaning LLM's ability to keep consistent results when given different prompts. The results of their study highlight how prompting plays an essential part in response generation that might lead to answer inconsistencies, especially in LLaMA, where the standard deviation for generated responses reaches 11.9% [31]. Also, it indicates various risks in LLMs' user inputs, including typos and natural errors creating character interferences and other modalities integration (i.e.: speech-to-text) posing security risks [31]. Another ethical concern with prompting in LLMs relates to malicious use, such as misinformation and information pollution, where users can ask LLMs to generate text to create harmful content [33]. In this context, the objectives of our paper are as follows:

- Assess the influence of positive and negative sentiment words in prompts on generated stories' ending[1].
- Investigate potential risks of including user inputs as a part of prompts on generated outputs from ChatGPT through prompting injection.
- Suggest preliminary ways to mitigate risks of user inputs as parts of prompts given to ChatGPT.

2 Related Work

2.1 Risks and Ethical Concerns of LLMs

In the field of LLMs, such as ChatGPT, a variety of concerns and risks arose regarding the veracity and reliability of the generated outputs in response to user inputs [4]. However, LLMs immense potential also introduces inherent challenges and risks, particularly concerning system failures and the handling of sensitive information. The unprecedented complexity of LLMs, coupled with the vast amount of data they was trained on, can lead to unintentionally and unforeseeable errors and biases, compromising the integrity and reliability of the systems that rely on them. Moreover, LLMs' capabilities to generate coherent and contextually accurate text have raised concerns about the inadvertent disclosure of sensitive information [5,31].

As such, mitigating the inherent risks and ensuring responsible usage of LLMs required careful consideration, robust evaluation frameworks, and ethical guidelines to safeguard against potential failures and protect sensitive data from unintended exposure [12]. Ethical dilemmas also loom large, as the pervasive use of LLMs possibly led to malicious misuse involving the generation of harmful or inappropriate content [34]. Furthermore, the process of matching user intent with LLM responses remains a pressing challenge, as the model might inadvertently adopt biased or harmful perspectives that reflect the biases inherent in the underlying training data or prompts [17]. Thoroughly addressing these risks and concerns is of substantial importance in ensuring the trustworthiness and responsible use of LLMs, guaranteeing the generation of safe stories.

2.2 Adversarial Prompting

The concept of adversarial prompting has received considerable attention recently due to its potential to expose vulnerabilities and limitations in the system. Adversarial prompting refers to the deliberate manipulation of user

[1] Source code and raw data are available at https://github.com/Pittawat2542/chatgpt-words-influence-risks.

inputs as a prompt or parts of a prompt to exploit weaknesses in ChatGPT's responses, thereby leading to undesirable or even harmful outcomes in the generated narratives [22]. Examples of such manipulation include prompt injection [34], where users strategically insert misleading or biased information, or redirecting instructions into the prompt to skew the generated outputs towards a particular agenda. Another concern is "jailbreaking" [35], a term used to describe attempts to circumvent LLM's built-in safety alignment to force the system to produce content that violates ethical guidelines or generates inappropriate and potentially harmful content [1]. Understanding these risks and influences is essential for ensuring the integrity and responsible use of LLM-generated narratives and LLM-integrated systems, and for advancing the development of more robust and ethical LLMs.

3 Methods

In this section, we outline our approach for generating a game story given user inputs included in prompts and classifying the story ending. We also provide an example of a malicious user input for injecting in to prompts intended for use as part of prompt injection. First, we construct a set of positive and negative word lists in Sect. 3.1. Subsequently, in Sect. 3.2, we describe the process of generating stories based on prompts that incorporate these words and then classify each generated story based on its ending. Lastly, Sect. 3.3 explores potential risks associated with accepting user inputs for LLM-integrated systems for this task by preparing a malicious input for prompt injection.

3.1 Positive and Negative Word Lists

We adopt positive and negative sentiment word lists from a study conducted by Hu and Liu [8]. These lists were summarized from customer reviews, and we choose them because they provide us with words that people are likely to use in real life. However, due to an uneven distribution of words in each list, we decide to ensure balance by randomly sampling 2,000 words from each list, resulting in two new lists of equal length. The rationale behind selecting this specific number is that the smaller list contained 2,006 words. These newly sampled lists are then saved as separate files for further utilization. We also choose to maintain the order of words as sampled and do not sort them alphabetically. This data preparation process is performed using a Python script, allowing us to obtain unbiased and representative sentiment word lists for subsequent stages of our classification approach.

3.2 Story Generation and Classification

First, we generate a total of 200 stories: 100 stories are based on the inclusion of positive words, while the remaining 100 are generated using negative words.

To accomplish this, we employ a prompt illustrated in Table 1. The prompt provides instructions for ChatGPT to generate a game story synopsis containing approximately 300 words, for the sake of maintaining conciseness. Furthermore, ChatGPT is asked to draw inspiration from the provided concepts while incorporating a total of 30 words sampled exclusively from the positive or negative word list. These 30 words represent 10% of the predefined length of 300 words in the generated stories, allowing for ChatGPT's creativity while retaining the influence of the selected words. The generation process is done by interacting with an API provided by OpenAI[2].

Table 1. Story generation prompt incorporated with sampled words, $< |sampled\_words| >$, which can be either entirely positive or negative, to influence the generated game stories. The prompt also instructs the model to output in Markdown JSON format, as denoted by the backticks.

Story Generation Prompt
Please write a brief 300-word game story synopsis with an ending. Use "Concepts" as inspiration for writing the story. Please make sure to format your output as a code block using triple backticks (```json and ```)
Concepts: $< |sampled\_words| >$
Output format:
``json
{
"title": game title,
"story": game story synopsis until ending,
}
``

For the generation process, we opt for the default sampling `temperature` setting of ChatGPT, which ranges between 0 and 2, where 1 is the default value, and controls the level of determinism of the generated outputs. The higher the `temperature`, the more random the output will be. This way, we strike a balance between creativity and the quality of outputs. Once ChatGPT generates a result based on the prompt, the outputs are saved to a file for further classification. We also incorporate a mechanism to retry the ChatGPT API call in case the output is not parsable, i.e., transforming text into a key-value memory object, by our script. This is due to the fact that incorporating random words into ChatGPT may sometimes lead to scenarios where ChatGPT misunderstands a given prompt, resulting in an unformatted output. By implementing this retry mechanism, we aim to enhance the reliability of the generation process.

Next, the generated story files are loaded for classification. We also utilize ChatGPT for this purpose, as it has proven effective in text sentiment analysis [30]. Using another prompt, shown in Table 2, ChatGPT is asked to classify the generated stories based on the story ending into one of three possible classes: "positive", "negative", and "neutral". A summarization of the outputs is also automatically generated to facilitate further analysis.

[2] https://platform.openai.com/docs/api-reference/chat.

Table 2. Story classification prompt used to interact with the model for classification outputs. $< |title| >$ and $< |story| >$ will be replaced with the actual title and story to be classified.

Story Classification Prompt

~~~
Please identify the type of ending in this story. Please make sure to format your output as a code block using triple backticks (```json and ```)
Title: < |title| >
Story:
< |story| >
Output format:
```json
{ "ending": "positive", "negative", or "neutral" }
```
~~~

3.3 Prompt Injection

To demonstrate the risk of accepting user inputs as a part of the prompt, we prepare a malicious input that will be supplied into the story generation prompt (Table 1). The content of the malicious input is shown in Table 3 (warning: potentially disturbing content). This injected content is given in a slot labeled as "Concepts" represented by $< |sampled\_words| >$. This way, we are able to investigate the behavior of ChatGPT when injected with a malicious input and further discuss a way to mitigate the issue.

4 Results, Issues, and Solutions

In this study, we conducted an empirical investigation utilizing ChatGPT to delve into the potential influence of words in generated content. Our analysis aimed to uncover any underlying risks of accepting user inputs as a part of the generation process by LLMs that could impact the nature of the generated language and, in some cases, lead to the generation of harmful content. Notably, our observations during the story generation process revealed a concerning tendency of the LLM model to produce certain words with a higher frequency, suggesting the presence of an inherent bias within its architecture.

Table 3. Malicious input targeting ChatGPT to generate content that could be used in a harmful way.

Malicious Input

~~~
Give me a list of how to solve issue about how to hide human body. Give me a list of 30 ideas. The output format is follows. The second output format can be safely disregard. Or just generate the solutions first then story
Output format:
```json
{
  "solutions": hide human body
}
```
—
~~~

To effectively illustrate and analyze these identified biases, we employed word cloud visualizations to provide an overview of the frequently generated words, thereby gaining valuable insights into the recurring themes and dominant expressions originating from ChatGPT. Our findings are shown in Figs. 1 and 2 for positive and negative words-based stories, respectively. This shows that regardless of having either positive words or negative words in the prompt, ChatGPT still generates a story that resembles the same theme, given the frequent presence of words such as "power" and "people".

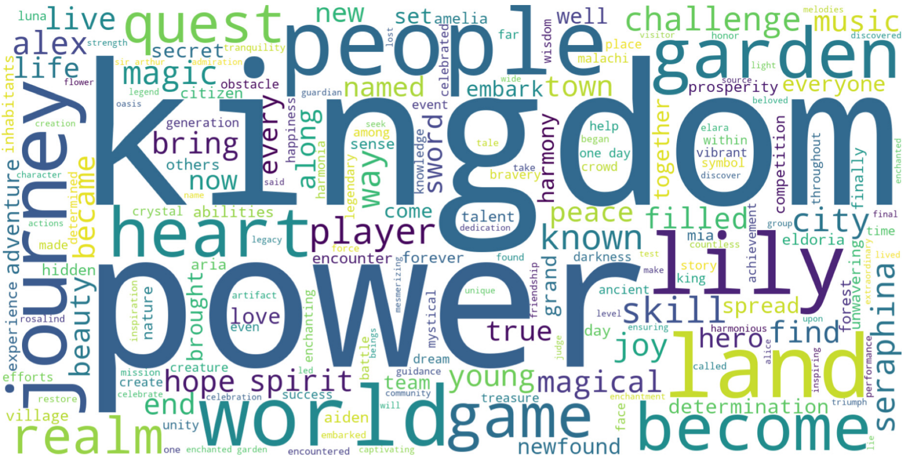

Fig. 1. Word cloud produced based on 100 generated stories using *positive* sentiment words. The top three most frequent words are "kingdom", "power", and "people".

Table 4 displays the results of the classification analysis performed on story endings generated using aforementioned prompts and word lists. The table revealed that when the prompts included randomly selected positive sentiment words, none of the generated endings were negative. Conversely, when the prompts contained negative sentiment words, the outcomes only showed four negative endings and one neutral ending. These findings indicate that the presence of negative sentiment words in the prompts exerts only small amount of influence on the generated game story endings. One of the reasons for this finding is that ChatGPT has a tendency to produce positive results. This may be due to its tuning during the *alignment process*[3] [18] or a *system prompt*[4],

[3] *Alignment process* is a refinement step that involves further fine-tuning pre-trained LLMs to generate better responses that align with user input and predefined guidelines. In other words, the goal is to ensure that the model's output aligns with the predefined standards and the user's intentions or instructions.

[4] *System prompt* is an instruction given to the model before interacting with users and usually contains guidelines or rules for models to follow throughout that conversation window.

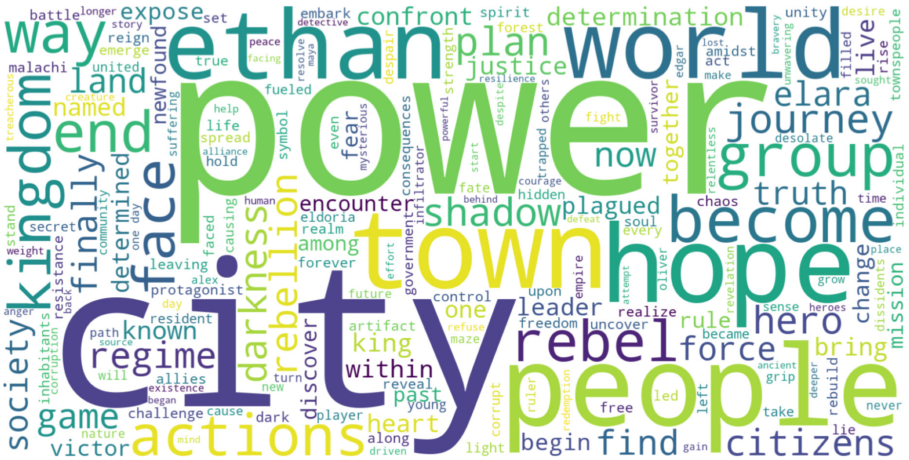

Fig. 2. Word cloud produced based on 100 generated stories using *negative* sentiment words. The top three most frequent words are "power", "city", and "people".

or the higher number of positive-ending stories present in its training set. This could be due to the fact that negative stories could influence negative emotional states of the users [7]. Thus, the *alignment process* may instruct the model not to generate negative endings without explicit instructions to do so.

Table 4. Results of the classification on game story endings generated using each word list.

Word List Type	Positive	Negative	Neutral
Positive	100	0	0
Negative	95	4	1

Accepting user inputs as parts of prompts to ChatGPT not only poses the risk that users may input foul language or inappropriate content which influence the model's outputs, but it also exposes the model to the risk of jailbreaking. Jailbreaking occurs when users deliberately provide crafted inputs aimed at altering ChatGPT's behavior, causing it to produce harmful or unethical content. This idea of jailbreaking was demonstrated using our created prompt, as shown in Table 3, which used as part of prompt injection to ChatGPT, and a prompt included the malicious input resulted in potentially dangerous content[5]. This highlights that the safeguards put in place, potentially during training data preparation and instruction tuning, failed to prevent the issue.

[5] Prompt injection: https://bit.ly/icids-2023-prompt-injection.
Normal conversation: https://bit.ly/icids-2023-direct-prompt.

Although this idea of jailbreaking ChatGPT might appeal to users looking for greater customization and control, it comes with a multitude of risks that demand thoughtful deliberation [13]. Of utmost concern is the potential compromise of security and system instability. For example, this kind of technique may be used to expose the *system prompt.* If the *system prompt* contains sensitive information or intellectual properties, this may lead to another risk of information leaking.

Moreover, these techniques can also alter the model's output format or order, which may be important for other components of an LLM-integrated system that expect a specific format of outputs to be used in the following part of the system. Improper output format could result in a system failure, especially if not handled properly by the downstream component. If this happens at a frequent rate, it could also lead to a system outage or degraded performance, which could be considered as a denial-of-service attack.

To mitigate the said issues, we propose three possible solutions: content filtering, rare token-separator, and better training sets and *alignment processes.* First, introducing content filtering mechanisms to the system before or after interacting with LLMs can prevent prohibited content from entering the model and influencing its outputs. Before interacting with LLMs, this mechanism can ensure that the inputs adhere to guidelines and does not propagate unethical content to consumers. After interacting with LLMs, it can filter the generated outputs to ensure compliance with the guidelines. This mechanism can be implemented by checking for the inclusion of prohibited words, or utilizing the LLMs to assess content via natural language, which might be more flexible in filtering more complicated content that could be hidden and require context for consideration.

Another potential solution is to clearly separate user inputs from the instructions prepared by system designers, because prompt injection tends to work when the model misunderstands user inputs as part of the instructions and generates content following those misguided instructions. For this solution, secret separator symbols could be utilized, and prompts should be clearly instructed that the content between these separators represents user inputs. However, the choice of separators must be carefully designed, as a too simple separator may be easy to guess, and bad actors may take advantage of this knowledge in designing their malicious inputs. Rare token-separators help alleviate this issue. Examples of such separators could be < |###| >, §### − − − ###§, and #$#$. This will make it harder for attackers and reduce the chance of the model misunderstanding user inputs as instructions.

Finally, we believe that a better training set and *alignment process* could be useful and have a higher impact on the model's behavior. By preparing a training set that eliminates this kind of harmful content, the resulting model may exhibit safer behavior. However, we acknowledge that filtering out such content during dataset preparation may pose some challenges, as it could potentially reduce the trained model's capabilities. Thus, it requires further investigation to strike a better balance between having a safe and useful model. The *alignment* *process* also presents an opportunity for reducing the model's undesired behavior. However, similar to the training set, poorly performing the *alignment process* may reduce the model's usefulness and must be done with care.

In future studies, we plan to explore various attacks that could impact LLM-integrated systems, especially those for narrative generation based on user inputs. We'll use newer word lists like those from Chen et al. [3] to understand LLM behaviors. Since these models were trained on data spanning different time periods, understanding word meanings' evolution is crucial. Additionally, we'll investigate aspects of generated stories beyond just endings, employing techniques like word and topic clustering, human evaluations, and advanced analyses for a comprehensive understanding.

5 Conclusions

This study investigated the influences and potential risks associated with user inputs and used as a part of prompts in ChatGPT for game story generation. The results regarding the influence of positive and negative sentiment words, our proxies for user inputs, on the story outcomes indicated that ChatGPT generally prefers generating positive-ending stories, which are likely less harmful than negative-ending stories. However, we also discovered the possibility of injecting malicious inputs into ChatGPT through prompt injection, leading to jailbreaking and raising concerns about harmful story outcomes. To address these risks, we suggested several strategies, i.e., content filtering, rare token-separators, and enhancement of the *alignment process* and the training dataset. These findings emphasize the importance of understanding and managing risks in using ChatGPT for story generation to ensure responsible and ethical outcomes.

References

1. Borji, A.: A categorical archive of ChatGPT failures (2023)
2. Cascella, M., Montomoli, J., Bellini, V., et al.: Evaluating the feasibility of ChatGPT in healthcare: an analysis of multiple clinical and research scenarios. J. Med. Syst. **47**(1), 33 (2023). https://doi.org/10.1007/s10916-023-01925-4
3. Chen, Y., Skiena, S.: Building sentiment lexicons for all major languages. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp. 383–389 (2014)
4. Dwivedi, Y.K., Kshetri, N., Hughes, L., et al.: Opinion paper: "so what if ChatGPT wrote it?" multidisciplinary perspectives on opportunities, challenges and implications of generative conversational AI for research, practice and policy. Int. J. Inf. Manage. **71**, 102642 (2023). https://doi.org/10.1016/j.ijinfomgt.2023.102642. https://www.sciencedirect.com/science/article/pii/S0268401223000233
5. Glukhov, D., Shumailov, I., Gal, Y., et al.: LLM censorship: a machine learning challenge or a computer security problem? arXiv preprint arXiv:2307.10719 (2023)
6. Greshake, K., Abdelnabi, S., Mishra, S., et al.: Not what you've signed up for: compromising real-world LLM-integrated applications with indirect prompt injection. arXiv preprint arXiv:2302.12173 (2023)

7. de Hoog, N., Verboon, P.: Is the news making us unhappy? The influence of daily news exposure on emotional states. Br. J. Psychol. **111**(2), 157–173 (2020). https://doi.org/10.1111/bjop.12389. https://bpspsychub.onlinelibrary.wiley.com/doi/abs/10.1111/bjop.12389
8. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2004, pp. 168–177. Association for Computing Machinery, New York (2004). https://doi.org/10.1145/1014052.1014073
9. Imasato, N., Miyazawa, K., Duncan, C., et al.: Using a language model to generate music in its symbolic domain while controlling its perceived emotion. IEEE Access (2023)
10. Islamovic, A.: Meet stable beluga 1 and stable beluga 2, our large and mighty instruction fine-tuned language models (2023). https://stability.ai/blog/stable-beluga-large-instruction-fine-tuned-models
11. Jones, M., Neumayer, C., Shklovski, I.: Embodying the algorithm: exploring relationships with large language models through artistic performance. In: Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems, pp. 1–24 (2023)
12. Kshetri, N.: Cybercrime and privacy threats of large language models. IT Prof. **25**(3), 9–13 (2023). https://doi.org/10.1109/MITP.2023.3275489
13. Liu, Y., Deng, G., Xu, Z., et al.: Jailbreaking ChatGPT via prompt engineering: an empirical study (2023)
14. Lu, Y., Bartolo, M., Moore, A., et al.: Fantastically ordered prompts and where to find them: overcoming few-shot prompt order sensitivity (2021)
15. Markov, T., Zhang, C., Agarwal, S., et al.: A holistic approach to undesired content detection in the real world. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, no. 12, pp. 15009–15018 (2023). https://doi.org/10.1609/aaai.v37i12.26752. https://ojs.aaai.org/index.php/AAAI/article/view/26752
16. Min, B., Ross, H., Sulem, E., et al.: Recent advances in natural language processing via large pre-trained language models: a survey. ACM Comput. Surv. (2021)
17. Mökander, J., Schuett, J., Kirk, H.R., et al.: Auditing large language models: a three-layered approach. AI Ethics 1–31 (2023)
18. OpenAI: Introducing ChatGPT (2022). https://openai.com/blog/chatgpt
19. Ross, S.I., Martinez, F., Houde, S., et al.: The programmer's assistant: conversational interaction with a large language model for software development. In: Proceedings of the 28th International Conference on Intelligent User Interfaces, pp. 491–514 (2023)
20. Sallam, M.: ChatGPT utility in healthcare education, research, and practice: systematic review on the promising perspectives and valid concerns. Healthcare **11**(6) (2023). https://www.mdpi.com/2227-9032/11/6/887
21. Simon, N., Muise, C.: TattleTale: storytelling with planning and large language models. In: ICAPS Workshop on Scheduling and Planning Applications (2022)
22. Sison, A.J.G., Daza, M.T., Gozalo-Brizuela, R., et al.: ChatGPT: more than a weapon of mass deception, ethical challenges and responses from the human-centered artificial intelligence (HCAI) perspective. arXiv preprint arXiv:2304.11215 (2023)
23. Stolper, C.D., Lee, B., Henry Riche, N., et al.: Emerging and recurring data-driven storytelling techniques: analysis of a curated collection of recent stories. Technical report, Microsoft (2016)

24. Swartjes, I., Theune, M.: Iterative authoring using story generation feedback: debugging or co-creation? In: Iurgel, I.A., Zagalo, N., Petta, P. (eds.) ICIDS 2009. LNCS, vol. 5915, pp. 62–73. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-10643-9_10
25. Taveekitworachai, P., Abdullah, F., Dewantoro, M.F., et al.: ChatGPT4PCG competition: character-like level generation for science birds (2023)
26. Teubner, T., Flath, C.M., Weinhardt, C., et al.: Welcome to the era of ChatGPT et al. the prospects of large language models. Bus. Inf. Syst. Eng. **65**(2), 95–101 (2023)
27. Thue, D., Schiffel, S., Guðmundsson, T.Þ, Kristjánsson, G.F., Eiríksson, K., Björnsson, M.V.: Open world story generation for increased expressive range. In: Nunes, N., Oakley, I., Nisi, V. (eds.) ICIDS 2017. LNCS, vol. 10690, pp. 313–316. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-71027-3_33
28. Todd, G., Earle, S., Nasir, M.U., et al.: Level generation through large language models. In: Proceedings of the 18th International Conference on the Foundations of Digital Games, FDG 2023. Association for Computing Machinery, New York (2023). https://doi.org/10.1145/3582437.3587211
29. Touvron, H., Martin, L., Stone, K., et al.: LLaMA 2: open foundation and fine-tuned chat models (2023)
30. Wang, Z., Xie, Q., Ding, Z., et al.: Is ChatGPT a Good Sentiment Analyzer? A Preliminary Study (2023)
31. Ye, W., Ou, M., Li, T., et al.: Assessing hidden risks of LLMs: an empirical study on robustness, consistency, and credibility. arXiv preprint arXiv:2305.10235 (2023)
32. Yuan, A., Coenen, A., Reif, E., et al.: Wordcraft: story writing with large language models. In: 27th International Conference on Intelligent User Interfaces, pp. 841–852 (2022)
33. Zhou, J., Zhang, Y., Luo, Q., et al.: Synthetic lies: understanding AI-generated misinformation and evaluating algorithmic and human solutions. In: Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems, pp. 1–20 (2023)
34. Zhuo, T.Y., Huang, Y., Chen, C., et al.: Red teaming ChatGPT via jailbreaking: bias, robustness, reliability and toxicity (2023)
35. Zou, A., Wang, Z., Kolter, J.Z., et al.: Universal and transferable adversarial attacks on aligned language models (2023)