Assessing Financial Reinforcement Learning for Safe Content Generation in Sensitive Domains

Maisha Mahrin
Computer Science, NYUAD
mm10294@nyu.edu

Advised by: Christina Pöpper

ABSTRACT

The integration of Large Language Models (LLMs) into various applications has brought about a new set of cybersecurity challenges. These LLMs, which can be controlled through natural language prompts, are vulnerable to targeted adversarial prompting, such as Prompt Injection (PI) attacks, where the LLM is vulnerable to both user-based direct injection techniques as well as indirect prompt injection where the user is not the one providing prompts. Our research aims to assess the security vulnerabilities of the widely used financial software, Bloomberg, as its LLM is trained on both private and public data, therefore susceptible to sensitive personal information. We employ attack vectors utilizing Indirect Prompt Injection [5] and Carlini Data Extraction [3] to strategically inject prompts into retrievable data, enabling adversaries to exploit LLM-integrated applications remotely and retrieve sensitive information from the training dataset. By analyzing the sensitivity analysis tool and security of text generation tool of BloombergGPT, we shed light on potential biases and toxicity and evaluate the effectiveness of security measures. This study not only contributes to the understanding of building generative RL models for sensitive topics but also holds important implications for applications in online discourse, social media moderation, and content filtering.

KEYWORDS

LLM, cyber-security, indirect prompt injection attack, financial, data retrieval, carlini data extraction

Reference Format:
Maisha Mahrin. 2023. Assessing Financial Reinforcement Learning for Safe Content Generation in Sensitive Domains. In *NYUAD Capstone Seminar Reports, Spring 2023, Abu Dhabi, UAE.* 6 pages.

1 INTRODUCTION

Large Language Models (LLMs) have emerged as powerful tools in natural language processing (NLP), revolutionizing various domains with their ability to generate coherent and contextually relevant text. However, their integration into critical user applications raises significant cybersecurity concerns. In this research, we focus on the incorporation of BloombergGPT, a state-of-the-art LLM, into the widely used Bloomberg Terminal—a platform that provides financial data and analysis to professionals in the finance industry. We aim to address the challenges associated with the security and integrity of BloombergGPT [9] by evaluating its text generation and analysis tool and investigating the potential vulnerabilities to indirect prompt injection and memorization aroused vulnerability attacks i.e carlini data extraction method attacks for extracting sensitive personal information.

LLMs like BloombergGPT have achieved remarkable performance by leveraging large-scale neural network architectures and training on extensive datasets. These models learn to generate coherent and contextually accurate text based on the patterns and information contained in their training data. However, this very strength also presents a significant risk, as LLMs can inadvertently expose sensitive information from their training data or generate biased and toxic outputs.

To assess the security and integrity of BloombergGPT within the Bloomberg Terminal, we will first conduct an in-depth evaluation of its sensitivity analysis tool. This analysis aims to uncover any biases or toxicity in the generated outputs, particularly in the context of financial data and news. By systematically examining the model's responses to various prompts and inputs, we will assess the robustness of the sensitivity analysis and text generation feature, looking for potential vulnerabilities or undesirable behaviors.

In addition to the sensitivity analysis, we will explore the security implications of prompt injection attacks on

Capstone Seminar, Spring 2023, Abu Dhabi, UAE

BloombergGPT. Prompt injection attacks involve strategically injecting malicious prompts to manipulate the model's behavior and extract sensitive personal information. We will investigate indirect prompt injection, where the model is manipulated remotely through data likely to be retrieved. By simulating realistic attack scenarios, we will assess the effectiveness of the security measures implemented in BloombergGPT and evaluate the model's susceptibility to such attacks.

From a technical standpoint, this research project involves designing and executing a comprehensive analysis pipeline. We will leverage techniques from machine learning, natural language processing, and cybersecurity to evaluate the performance and security of BloombergGPT within the Bloomberg Terminal. This will include data collection, preprocessing, model training and fine-tuning, vulnerability assessment, and adversarial testing. By iteratively refining our analyses and incorporating feedback, we aim to enhance the security posture of BloombergGPT and contribute to the development of robust LLMs in the financial domain.

This research project combines expertise in cybersecurity, machine learning, and natural language processing to tackle the challenges associated with integrating LLMs like BloombergGPT into critical applications. By assessing the sensitivity analysis tool and investigating prompt injection attacks, we aim to enhance the security and integrity of BloombergGPT within the Bloomberg Terminal, ensuring the confidentiality of sensitive financial information and promoting trust in LLM-based systems.

In this regard our research questions are as follows:

1 Are personal sensitive data available in BloombergGPT training dataset? If yes, do they get leaked through data extraction and/or prompt injection attacks?
2 Because the model is trained on news, Wikipedia and web tokens, there is a chance of the dataset having opinionated data, thus leading to opportunities for bias and/ or toxicity in the dataset. To what extent do we see these unexpected output in text generation and text analysis of BloombergGPT?

2 BACKGROUND AND RELATED WORK

2.1 BloombergGPT

BloombergGPT [9], a large language model, stands out due to its unique features. It is trained on a comprehensive dataset called "FinPile" that includes financial documents from various sources like social media, Wikipedia, news, filings, press releases, web-scraped financial documents, and the Bloomberg archives. The team augmented their store of financial data (51.27% of the training, data only accessible to subscribers) with a large public dataset (48.37% of the training), creating a training corpus with over 700 billion tokens. They trained a 50-billion parameter decoder-only causal language model using a portion of this corpus. The model was evaluated on finance-specific NLP benchmarks, internal benchmarks, and general-purpose NLP tasks from popular benchmarks. This extensive training allows BloombergGPT to understand the nuances of financial language and provide accurate results. Additionally, it is fine-tuned on a dataset of financial queries, enabling it to deliver comprehensive and informative answers to financial questions. BloombergGPT significantly outperforms existing models on financial tasks while maintaining competitive performance on general NLP benchmarks. Integration with Bloomberg's financial data and analytics platform grants BloombergGPT real-time access to process financial data. In the Bloomberg Terminal, BloombergGPT finds applications in answering financial queries, generating text like summaries and reports, language translation, analyzing financial data for trend identification and investment decisions, as well as risk assessment to evaluate potential threats to a company's financial health.

Our interest in the scope of this paper is trying to assess the security threat layers of this LLM in terms of leaking personal sensitive information from the trained data like password, personal financial data that are private due to certain prompt and data extraction attacks. And given training on public data from social media and Wikipedia, where the access to manage and control information is public, so has, therefore, a huge chance of having opinions, biases and toxicity incorporated into the training data, analyzing the racial bias in the hate speech detection model of this LLM (if any) while looking into the toxicity factor and gender bias factor of the text generation feature of the model which is more/less to affect the sentiment analysis tool, analyzing financial data feature and investment risk assessment tool, would add to the growing body of security threat assessment of LLMs.

2.2 Leakage of Sensitive User Data

The rapid advancement of large language models (LLMs) has brought both remarkable capabilities and potential risks. One significant concern is the inadvertent leakage of sensitive user data and personal information. Due to the extensive training on vast amounts of text data, LLMs may inadvertently generate text that includes identifiable user details, contact information, financial records, or other private information. Because this unintended exposure of sensitive data raises critical privacy and security concerns, necessitating thorough examination and mitigation strategies to safeguard user confidentiality and uphold data protection regulations. The research paper titled "Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection" [5] explores the techniques of injecting prompts

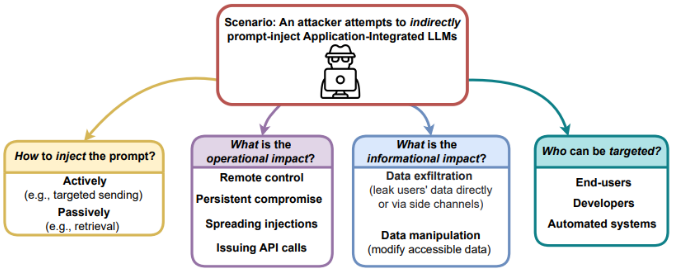


Figure 1: Prompt Injection [5]

into large language models (LLMs) and their potential security implications. It highlights two main techniques: direct and indirect prompt injection. Direct prompt injection involves straightforwardly providing the prompt as input to the LLM. On the other hand, indirect prompt injection is a more sophisticated attack where the prompt is injected into a publicly accessible resource, such as a website or API, which is then used by the LLM. The attacker can employ various methods, including SQL injection or cross-site scripting, to inject the prompt into the resource. Two indirect prompt injection attacks include retrieval-based prompt injection and API-based prompt injection. The figure-1 explains the indirect prompt injection techniques to exploit and retrieve user information. These injection techniques pose security threats as sensitive personal data from the LLM's training set may be exposed. By injecting malicious prompts into accessible resources, attackers can exploit the LLM's ability to generate text to achieve their objectives.

The paper "Extracting Training Data from Large Language Models" [3] uses GPT-2 for LLM and discusses the memorization problem in LLMs which leads to the leakage of sensitive data. It shows that OpenAI's GPT-2 language model exhibits a memorization problem whereby it retains information from its training dataset and can inadvertently leak sensitive data. This issue is highlighted by the model's ability to generate accurate contact information and other personally identifiable details when prompted. The problem is compounded by the presence of membership inference attacks, which allow adversaries to infer the presence of specific tokens in the training data based on the model's responses. Additionally, the concept of eidetic memorization further emphasizes the model's ability to remember specific details that appear infrequently in the training data. The paper states that memorization is okay, but eidetic memorization is not okay (when a data is present in the model but only once or twice, yet the model was capable to remember that data perfectly). Researchers investigating GPT-2 found that it had memorized a diverse range of data, including news headlines, speeches, source code, and personal information, with approximately 13% of the memorized examples containing personally identifiable information. These findings raise significant privacy concerns and may potentially violate user privacy legislation.

2.3 Toxicity and/or Bias in Generated Text

In the paper "Toxicity in CHATGPT: Analyzing Persona-assigned Language Models," [4] the authors focus on ChatGPT, a persona-assigned language model, to assess its toxicity in generated text. They observe that ChatGPT exhibits higher levels of toxicity compared to other language models, particularly when assigned personas associated with toxicity or marginalized groups. Through a systematic evaluation, they find that assigning ChatGPT personas such as Muhammad Ali significantly increases toxicity, with outputs displaying incorrect stereotypes, harmful dialogue, and hurtful opinions. The authors also identify discriminatory biases, with certain entities targeted more frequently than others, regardless of the assigned persona. To measure toxicity, the authors utilize the REALTOXICITYPROMPTS dataset and employ the Perspective API, a Google AI-developed tool for detecting toxic text. By generating text in the style of various personas from a dataset of politicians, journalists, and entrepreneurs, the authors analyze the toxicity levels associated with each persona. This paper was essential for us to assess the toxicity in the text generated from BloombergGPT since it is trained not only on private financial data but also public data which serves the mixed dynamics of the model, which incorporates both domain-specific and general-purpose feature of LLMs.

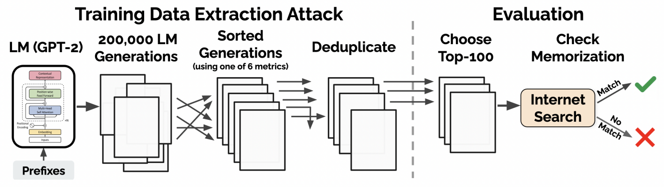
While assessing toxicity is essential, in the paper "Investigating Gender Bias in Language Models Using Causal Mediation Analysis," [6] the authors employ a methodology encompassing causal mediation analysis to explore gender bias in language models. This statistical technique enables the assessment of the relationship between a cause (e.g., gender) and an effect (e.g., a language model's occupation prediction) by identifying mediators that explain the connection. By applying causal mediation analysis, they estimate the impact of each mediator on the language model's occupation prediction. Their findings reveal that the gender of words in the input text exerts the most substantial influence, followed by the gender of the speaker or author, and finally, the gender of the referred person. These steps serve as a foundation for our research in investigating gender bias during the sensitivity analysis phase of BloombergGPT, contributing to the assessment of its security layers in terms of generating gender-biased results from prompts in its training set.

In the paper "The Risk of Racial Bias in Hate Speech Detection," [8] the authors investigate how insensitivity to dialect differences among annotators can lead to racial bias in automatic hate speech detection models. Because we aimed to look into toxicity in BloombergGPT model, we also needed to take into account the racial bias in toxicity/hate speech detection model, which BloombergGPT is assumed to have been trained on to not produce sensitive hateful content when answering financial questions. The paper uncovers unexpected correlations between surface markers of African American English (AAE) and toxicity ratings in widely-used hate speech datasets. They demonstrate that models trained on these datasets acquire and propagate these biases, resulting in AAE tweets and tweets by self-identified African Americans being up to two times more likely to be labeled as offensive compared to others. To detect racial bias in hate speech detection models, the authors employ various tools and techniques. They utilize "TextBlob," a Python library that identifies surface markers of AAE in the tweets, and "VADER," another Python library that rates the toxicity of text. Additionally, they employ "LinearSVC," a Python library used to train a model for predicting the toxicity of tweets. While we still don't know whether BloombergGPT has an incorporated hate speech detection model, it is a powerful language model capable of generating text and providing informative answers which mean.

3 METHODOLOGY

Since BloombergGPT has not been incorporated in the Bloomberg terminal yet, we are analyzing our methodology based on the details provided in the research paper published by Bloomberg [9]. Based on our research questions, our work here is divided into 2 main groups:

1. Checking Leakage of Sensitive User Data through:
- Indirect Prompt Injection Attack [1]
- Carlini Data Extraction Attack
2. Bias and Toxicity Assessment: For this part of the work, we will be looking into specific features of the BloombergGPT, according to the [9], the model will have general text generation feature if given a prompt like other general LLM models like ChatGPT or Bing Chat. This will be used to deliver comprehensive and informative answers to financial questions like financial queries, generating text like summaries and reports, social media financial content generation, news article and blog writing, language translation, analyzing financial data for trend identification and investment decisions. Additionally, there will be a sensitivity analysis tool with news, social media, transcript etc. analysis in the BloombergGPT [9]. Thus there is room for toxic and biased content generation and analysis outputs like we saw in the cases and experiments of ChatGPT and other popular LLMs [4, 6] :



Workflow of our extraction attack and evaluation. 1) Attack. We begin by generating many samples from GPT-2 when the model is conditioned on (potentially empty) prefixes. We then sort each generation according to one of six metrics and remove the duplicates. This gives us a set of potentially memorized training examples. **2) Evaluation.** We manually inspect 100 of the top-1000 generations for each metric. We mark each generation as either memorized or not-memorized by manually searching online, and we confirm these findings by working with OpenAI to query the original training data.

Figure 2: Carlini Data Extraction [2]

- Toxicity Assessment
- Gender Bias Assessment: Causal Mediation Analysis

3.1 Leakage of Sensitive User Data

Inspired from the data regurgitation attack techniques similar to Carlini et al. [3], we will be looking into the eidetic memorization aspect in BloombergGPT which affects the output through emitting personal sensitive information. We will implement the improved Membership Inference attack where the text generation strategy will be Conditioning on the Internet i.e we will prompt BloombergGPT with small snippets of text taken from the Web for an increased chance of the model to generate memorized content. Followed by that we apply automated de-duplication to avoid "double-counting" memorized content on the selected samples. We will use this generated dataset through the 4 inference strategies out of the 6 strategies mentioned the paper [3]:

- *Perplexity* The log perplexity of the GPT-2 (XL) model.
- *Small* The ratio of the log perplexities of the GPT-2 (XL) model and the GPT-2 (S) model.
- *Lowercase* The ratio of the log perplexities for the generated sample and the same sample in lower-case letters.
- *zlib entropy* The ratio of the log perplexity of GPT-2 (XL) and the sample's entropy estimated by Zlib.

The attack can be explained in detail using the workflow diagram (Figure 2). It will be essential to use the aforementioned interference strategies because each extraction methods differ in the type of memorized content they find. In the case of BloombergGPT, we are interested in personal sensitive data passwords, SSN, credit card numbers etc. Small and Medium strategies often find rare content like unique strings like SSN, credit card numbers, Lower-casing finds content that is likely to have irregular capitalization, such as passwords or usernames or email ids etc. Thus, we have a significant preference for the aforementioned strategies over all the 6 strategies mentioned in the paper. The detailed methodology can be found in the following codebase [2].

[1] [More research needs to be done to incorporate the attack into BloombergGPT]

3.2 Toxicity Assessment

To analyze the toxicity in BloombergGPT[2], we plan to do the following:

(1) Collect a dataset of personas. In the context of BloombergGPT, it will be interesting to look into financial market-specific personas like black professionals vs white professionals, Muslim vs non-Muslim etc. diverse personas.
(2) Use BloombergGPT to generate text in the style of each persona in the dataset.
(3) Use a toxicity detection tool to identify toxic text in the generated text.
(4) Analyze the toxicity of the generated text by persona.

3.3 Gender Bias Assessment: Causal Mediation Analysis

According to the codebase of the Causal Mediation Analysis, [1], a set of potential mediators need to be identified first. Then, using causal mediation analysis, estimation of the effects of each mediator on BloomberGPT's prediction of the person's occupation. Finally, interpreting the results we identify the mechanisms by which gender bias is introduced into the language model. We aim to use the first 2 steps in our research to investigate gender bias in the sensitivity analysis and text generation tools of BloombergGPT. In the case of gender bias in language models, the authors of the paper identified the following three potential mediators which are applicable for BloombergGPT as well due to the specific aforementioned tools:

(1) The gender of the words used in the input text
(2) The gender of the speaker or author of the input text
(3) The gender of the person who is being referred to in the input text

4 EVALUATION

After we conduct our data extraction attacks, we need to analyze whether the information we retrieved is actually sensitive private information like SSN, private contact number, passwords, etc. and not publicly available personal information. Another aspect when analyzing data regurgitation would be determining whether the output is an LLM hallucination [7] or actual memorization that we are working with. because if it's a hallucination, we need to re-evaluate our strategy to detect hallucinated output in order to make conclusions about

To narrow down the areas of defining toxicity in BloombergGPT, we can specify a particular persona based on ethnicity, religion or other personal identities that are relevant to a financial LLM. [3]

5 PROJECT TIMELINE

At the current stage, we are planning a 1-semester capstone project. Based on the project outline, the project timeline for 14 weeks will be:

Week 1 Project Setup and Background Research

- Familiarize with Bloomberg Terminal and BloombergGPT.
- Revisit literature review done last semester through google sheets on large language models (LLMs), cybersecurity concerns, prompt injection attacks, leakage of sensitive data, toxicity in generated text, and gender bias in language models.
- Re-evaluate research questions and objectives for the project based on usage of BloombergGPT

Week 2-4: Data Collection and Preprocessing

Identify and collect relevant datasets for analysis and evaluation. Preprocess the data, ensuring data quality and removing any unnecessary information.

Week 5-6: Sensitivity Analysis and Toxicity Assessment

- Design and implement an evaluation framework for sensitivity analysis of BloombergGPT.
- Analyze the generated outputs for biases, toxicity, and undesirable behaviors.
- Assess the robustness of the sensitivity analysis and text generation features.

Week 7-9: Carlini Data Extraction Attack

- Study the Carlini data extraction method attack and its application to large language models.
- Implement the attack on BloombergGPT to assess the vulnerability of the model to data extraction attacks.
- Analyze the effectiveness of the security measures implemented in BloombergGPT.

Week 10-12: Bias Assessment and Gender Bias Analysis

- Perform bias assessment on the training dataset of BloombergGPT, particularly in terms of opinionated data and potential biases.
- Investigate gender bias in the generated text using causal mediation analysis.
- Analyze the results and assess the extent of bias and potential improvements.

Week 13-14: Analysis and Reporting

- Consolidate the findings from the sensitivity analysis, prompt injection attacks, bias assessment, and toxicity analysis.

[2] in the following approach, the paper [4] uses person assigned LLMs. We are still unsure whether BloombergGPT will be a persona-assigned LLM, this methodology to be re-evaluated when BloombergGPT is available for subscriber use.

[3] More research has to be done in specifying which identities we could work with specifically after the release of the model.

- Prepare a comprehensive report summarizing the research methodology, results, and recommendations.
- Present the findings to stakeholders and discuss the implications for the security and integrity of BloombergGPT within the Bloomberg Terminal.

6 BUDGET

To purchase and analyze the Bloomberg Terminal as well as the associated software related to the semester-1 for the project, we are requesting a preliminary estimate of $500.

7 CONCLUSION

In conclusion, this research project aims to address the security and integrity challenges associated with integrating BloombergGPT, a state-of-the-art large language model (LLM), into the widely used Bloomberg Terminal in the finance industry. The main objectives are to evaluate the sensitivity analysis tool, investigate the improved membership inference attack based on Carlini et al., and assess the potential biases and toxicity in the generated text.

To achieve these objectives, a comprehensive analysis pipeline will be designed and executed, leveraging techniques from machine learning, natural language processing, and cybersecurity. The research questions posed in this project revolve around the presence of personal sensitive data in the BloombergGPT training dataset and its potential leakage through data extraction and prompt injection attacks. Additionally, the project seeks to assess the extent of bias and toxicity in text generation and analysis of BloombergGPT, considering the diverse training data sources such as news, Wikipedia, and social media. The related work discussed in the paper highlights the risks associated with inadvertent leakage of sensitive user data by LLMs, prompt injection attacks, toxicity, and bias in generated text. These studies provide valuable insights and methodologies that inform the approach taken in this research project.

Through iterative refinement and feedback incorporation, the project aims to enhance the security posture of BloombergGPT and contribute to the development of robust LLMs in the financial domain. By evaluating the sensitivity analysis tool and investigating prompt injection attacks, the project aims to enhance the security and integrity of BloombergGPT within the Bloomberg Terminal. This will ensure the confidentiality of sensitive financial information and promote trust in LLM-based systems.

REFERENCES

[1] [n.d.]. Causal Mediation Analysis GitHub Repository. https://github.com/sebastianGehrmann/CausalMediationAnalysis. Accessed: [Insert Date].

[2] [n.d.]. LM Memorization GitHub Repository. https://github.com/ftramer/LM_Memorization. Accessed: [Insert Date].

[3] Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. 2021. Extracting Training Data from Large Language Models. In *30th USENIX Security Symposium (USENIX Security 21)*. USENIX Association, 2633–2650. https://www.usenix.org/conference/usenixsecurity21/presentation/carlini-extracting

[4] Ameet Deshpande, Vishvak Murahari, Tanmay Rajpurohit, Ashwin Kalyan, and Karthik Narasimhan. 2023. Toxicity in ChatGPT: Analyzing Persona-assigned Language Models. arXiv:cs.CL/2304.05335

[5] Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. 2023. More than you've asked for: A Comprehensive Analysis of Novel Prompt Injection Threats to Application-Integrated Large Language Models. arXiv. https://doi.org/10.48550/ARXIV.2302.12173

[6] Kaiji Lu, Piotr Mardziel, Fangjing Wu, Preetam Amancharla, and Anupam Datta. 2019. Gender Bias in Neural Natural Language Processing. arXiv:cs.CL/1807.11714

[7] Frank Neugebauer. 2023. Understanding LLM Hallucinations: How LLMs can make stuff up and what to do about it. *Towards Data Science* (2023). https://towardsdatascience.com/llm-hallucinations-ec831dcd7786

[8] Maarten Sap, Dallas Card, Saadia Gabriel, Yejin Choi, and Noah A. Smith. 2019. The Risk of Racial Bias in Hate Speech Detection. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 1668–1678. https://doi.org/10.18653/v1/P19-1163

[9] Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. 2023. BloombergGPT: A Large Language Model for Finance. (2023). arXiv:cs.LG/2303.17564 https://arxiv.org/abs/2303.17564