УДК 004.855.5

Разработка алгоритма обнаружения Prompt Injection с использованием метода дообучения LoRA

Свиридов Д.А. (Университет ИТМО)
Научный руководитель – кандидат технических наук, доцент Менщиков А.А.
(Университет ИТМО)

Введение. С развитием искусственного интеллекта и широким распространением языковых моделей в различных сферах жизни возникает новая угроза — Prompt Injection. Этот тип атаки заключается в манипулировании запросами, направляемыми на языковую модель, с целью получения нежелательных или вредоносных результатов. Для повышения защищенности программных продуктов основанных на использовании языковых моделей появляется необходимость в надежном решении по защите от атак Prompt Injection.

Одним из эффективных методов защиты от таких атак является дообучение языковых моделей. Наиболее распространенный метод дообучения “fine-tuning” [1] требует больших вычислительных ресурсов, поэтому в данной работе предлагается использовать метод LoRA (Low-Rank Adaptation) для повышения устойчивости модели в условиях применения Prompt Injection. LoRA представляет собой инновационный подход к дообучению моделей, который позволяет адаптировать их к специфическим задачам с минимальными затратами вычислительных ресурсов, что делает его подходящим для использования в реальных условиях [2].

Основная часть. Суть предложенного решения заключается в разработке алгоритма очистки запроса от Prompt Injection. Для этого были выделены следующие этапы разработки:

1. **Анализ и классификация атак**. В данном разделе были проанализированы наборы данных, содержащие Prompt Injection, а также работы приводящие свои классификации атак. На основе этого была составлена своя классификация существующих методов атак и выделены их основные особенности.

2. **Разработка алгоритма**. На основе классификации из предыдущего раздела был разработан алгоритм, который включает в себя наиболее эффективные методы обнаружения и очистки запроса от Prompt Injection. Для большинства атак был составлен набор данных для дообучения методом LoRA. Для обнаружения некоторых методов было решено использовать системный запрос, содержащий подсказки и примеры для обнаружения атаки.

3. **Эксперименты и результаты.** После составления набора данных и составления необходимых запросов, была написана программа, которая проводила очистку запросов. В результате мы провели эксперименты с и определили эффективность приведенных методов защиты.

Вывод. В результате проделанной работы были рассмотрены известные методы реализации атак Prompt Injection, а также составлена их классификация. На основании особенностей проанализированных методов было предложено программное решение, которое позволяет повысить защищённость программных продуктов от атак Prompt Injection.

Список использованных источников:

1. Alrashed M., Chen S., Piet.J, Sitawarin C. и др. Jatmo: Prompt injection defense by task-specific finetuning // In European Symposium on Research in Computer Security. – 2023 – №2 – С. 6–8. https://arxiv.org/pdf/2312.17673

2. Liu Y., Jia Y. и др. Formalizing and Benchmarking Prompt Injection Attacks and Defenses // 33rd USENIX Security Symposium – 2024 – №5 – С. 1833–1836. https://www.usenix.org/system/files/usenixsecurity24-liu-yupei.pdf
3. Chen S., Piet.J, Sitawarin C. и др. StruQ: Defending Against Prompt Injection with Structured Queries // 34rd USENIX Security Symposium – 2024 – №3 – С. 4–7. https://arxiv.org/pdf/2402.06363