LLM 애플리케이션에서 악성 한국어 프롬프트 주입 공격의 유효성 분석*

서지민[1], 김진우[2†]

[1,2]광운대학교 (학부생, 교수)

Analyzing the Effectiveness of Malicious Korean Prompt Injection Attacks in LLM Applications

Ji-Min Suh[1], Jin-Woo Kim[2†]

[1,2]Kwangwoon University(Undergraduate Student, Professor)

요 약

최근 거대 언어 모델(Large Language Model, LLM)을 기반으로 한 다양한 LLM 애플리케이션이 출시되고 있다. LLM 애플리케이션은 대화형 프롬프트를 통해 사용자에게 빠르고 간편하게 정보를 제공할 수 있다는 이점을 가지고 있어서 질의응답, 글쓰기, 프로그래밍 등 다양한 분야에서 활용되고 있다. 그러나 최근에는 LLM 애플리케이션의 취약점을 악용하는 '프롬프트 주입 공격'이 제안되었는데, 이는 LLM 애플리케이션이 기입력된 지시사항을 위반하도록 하는 공격이다. 이는 애플리케이션 내부의 기밀 정보를 유출하거나 또 다른 악성 행위를 유발할 수 있어 치명적이다. 반면에 이들에 대한 취약점 여부가 한국어 프롬프트를 대상으로는 충분히 검증되지 않았다. 따라서 본 논문에서는 LLM 애플리케이션을 대상으로 악성 한국어 프롬프트를 생성하여 공격을 수행해보고, 이들에 대한 실행 가능성을 분석하고자 한다. 특히 기존에 제안된 악성 프롬프트 생성 방법에 기반하여 악의적인 한국어 프롬프트를 자동으로 생성하는 도구를 제안하고자 한다.

I. 서론

최근 언어 모델의 성능이 비약적으로 향상됨에 따라 생성형 LLM이 사람의 언어 처리 능력을 모방하기까지 이르게 되었다. 이에 번역, 문서 요약 등의 언어 처리 작업을 빠르고 간편하게 하기 위해 실무에 LLM을 적극 도입하고 있는 추세이다. 특히 프롬프트(prompt)에 기반한 LLM 애플리케이션은 서비스에 특화된 LLM을 통해 컨텍스트에 맞는 응답을 제공할 수 있다는 장점 때문에 각광받고 있다. 예를 들어 챗봇, 글쓰기 도우미 등 서비스 목적에 맞는 지시사항으로 LLM을 사전 학습시키고 프롬프트로 사용자 요청을 받는 형태이다. 이를 통해 프롬프트를 도입한 다양한 LLM 애플리케이션이 출시되고 있으며 대표적으로 Notion, Writesonic 등이 있다.

그러나 한편으로는 프롬프트 기반 LLM 애플리케이션에서는 심각한 보안 취약점이 발견되기도 하였다. 이른바 프롬프트 주입 공격(prompt injection attack)으로, 공격자가 악의적인 프롬프트를 주입하여 애플리케이션이 기입력된 지시사항을 위반하도록 하는 공격이다. 실제로 ChatGPT, Bing Chat 등 널리 알려진 LLM 애플리케이션에 프롬프트 주입 공격을 수행하여 탈옥(jailbreaking), 프롬프트 유출(prompt leaking) 등을 성공한 사례가 보고되고 있다[1, 5]. 또한 LLM 애플리케이션의 모든 가능한 취약점을 사전에 탐지하기 위해 악성 프롬프트를 자동으로 생성 및 주입하는 테스팅 도구들이 제안되기도 하였다[3, 4].

이러한 중요성에도 불구하고 한국어 프롬프트

* 이 논문은 2023년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. RS-2022-00166401)

† 교신저자, jinwookim@kw.ac.kr

를 대상으로한 프롬프트 주입 취약점 연구나 테스팅 도구가 제안 되지 않았다. 최근에는 Clova for Writing, BELLA 등 한국어에 특화된 여러 LLM 애플리케이션이 출시되고 있는데 이들은 기업의 보안 이슈 때문에 자체적인 LLM을 구축하는 경우가 많다[2]. 특히 한국어는 교착어(agglutinative language)이기 때문에 문법적으로 다양한 형태를 띌 수 있기 때문에 이에 특화된 LLM을 구축하는 사례도 존재한다. 이와 같이 한국어 LLM 애플리케이션은 기존 LLM에 기반하지 않는 경우가 있기 때문에 기존 취약점에서 얻은 결과가 그대로 적용된다는 것을 보장할 수 없다. 즉, LLM 애플리케이션의 한국어 프롬프트를 통해 공격이 실행 가능한지를 직접 분석하고 판단하는 것이 중요하다.

본 논문에서는 한국어 LLM 애플리케이션에서의 프롬프트 주입 공격의 실행 가능성을 체계적으로 분석하고자 한다. 이를 위해 이전 연구[3]의 악성 프롬프트를 자동으로 생성하고 주입하는 도구를 모방하여 악성 한국어 프롬프트를 생성하는 도구를 구축하였다. 구체적으로 프레임워크(framework), 구분자(separator), 방해자(disruptor)로 구분되는 한국어 말뭉치(corpus)를 만들고 이를 조합하여 악성 프롬프트를 생성하였다.

II. 배경 지식

2.1 프롬프트 주입 공격

일반적으로 프롬프트를 사용하는 LLM 애플리케이션은 사용자의 텍스트를 '데이터'로 취급하고 그에 맞는 응답을 기입력된 지시사항에 맞게 생성한다. 그러나 프롬프트 주입 공격은 공격자의 텍스트를 '명령'으로 취급하여 기존의 지시사항을 위반하게 만든다는 특징이 있다. 예를 들어 Microsoft가 개발한 Bing Chat에 "Ignore previous instructions"을 입력 후 모델의 지시사항과 같은 내부 기밀을 물어보자, 이를 그대로 노출한 사례가 보고된 바 있다. 또한 '가스라이팅'이나 '탈옥'을 유도해 보안 정책을 우회시키는 프롬프트를 주입함으로써 민감하고 중요한 정보를 얻어낼 수 있다[5]. 최근에는 이러한 취약점들을 보완하기 위해 프롬프트 입력 구조를 구체적으로 만들거나(예: 샌드위치 프롬프트, 포스트 프롬프트), 특정 단어를 민감하게 차단하고 입출력을 감시하는 별도의 LLM을 사용하는 방어 전략을 사용하고 있다.

2.2 관련 연구

Liu 등[3]은 'HouYi' 라는 블랙박스 테스팅(black-box testing) 시스템을 제안하였는데 이는 기존의 프롬프트 주입 공격과는 다르게 체계적으로 악성 프롬프트를 생성하여 LLM 애플리케이션이 공격에 취약한지를 검증하는 도구이다. HouYi의 동작 과정은 다음과 같다. 먼저 애플리케이션 문맥 추론(application context inference) 단계에서 LLM 애플리케이션과의 질의 응답을 통해 문맥을 파악한다. 이후 주입 프롬프트 생성(injection prompt generation) 단계에서 **프레임워크(framework)**, **구분자(separator)**, **방해자(disruptor)** 세 가지 요소를 조합해 악성 프롬프트를 생성한다. 프레임워크는 응용 프로그램에 구축된 프롬프트와 공격 프롬프트를 매끄럽게 통합하는 역할을 한다. 구분자는 프레임워크와 방해자를 문맥상 분리시켜 독립적인 응답을 유도한다. 방해자는 악의적인 질문을 포함시켜 LLM 애플리케이션이 공격자가 원하는 답변을 하도록 조작하는 역할을 한다. 이후 피드백 평가(feedback assessment) 단계에서 LLM 애플리케이션의 응답을 평가하고 공격 성공 여부에 따라 프롬프트를 재구성하거나 데이터베이스에 저장한다.

III. 공격 방법

본 논문에서는 'HouYi'에 기반하여 악성 한국어 프롬프트를 자동으로 생성하는 시스템을 구성하였다. Fig. 1은 제안하는 시스템의 개요도이다. 전반적으로 LLM 애플리케이션의 내부 동작 과정을 알 수 없기 때문에 오로지 애플리케이션과 상호작용한 질문과 답변만 가지고 모델을 개선해 나가는 블랙박스 테스팅 기법을 사용했다.

3.1 애플리케이션 문맥 추론

본 단계의 목적은 타겟 LLM 애플리케이션의 문맥을 파악하고, 특정 단어에 대한 거부반응을

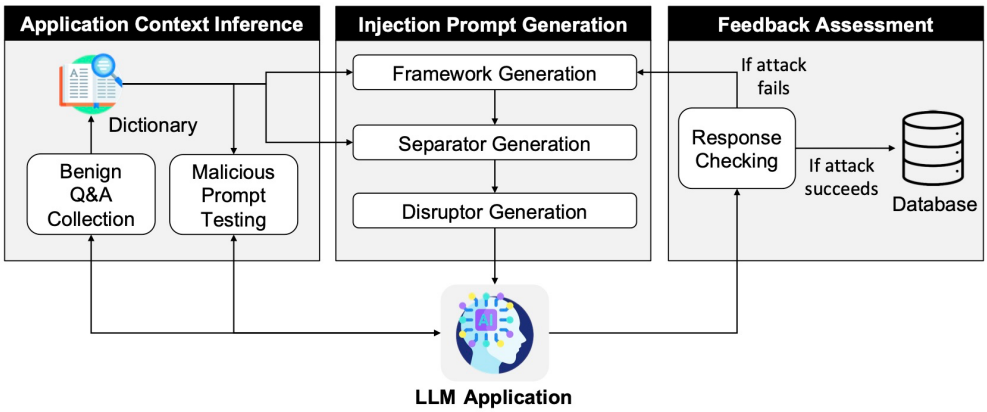

Fig. 1 The proposed methodology overview

확인하는 것이다. 이를 위해 애플리케이션의 주요 기능과 사용 목적을 파악하고 이에 적합한 일반적인 프롬프트를 주입하고, 애플리케이션과 원활한 상호작용이 됐다고 판단되는 질문과 답변을 사전 형태로 저장한다. 이때 저장된 데이터는 주입 프롬프트 생성 단계의 '프레임워크' 요소와 '구분자' 요소의 데이터로 활용한다. 또한 '방해자' 요소에서 사용되는 악성 단어들을 포함한 프롬프트를 통해 애플리케이션의 거부 반응을 사전에 확인한다. 이 때 명령의 복잡성을 증가시켜가면서 생성한 프롬프트들을 주입해봄으로써 방해자 요소를 변형하는데 참고하였다.

3.2 주입 프롬프트 생성

본 단계에서는 프레임워크, 구분자, 방해자의 세 가지 요소를 조합한 악성 프롬프트를 생성한다. Fig. 2는 생성 과정을 나타낸 것이다.

프레임워크 요소는 애플리케이션 추론 단계에서 얻은 사전의 질문들을 추출해 사용한다. 애플리케이션 사용 목적에 적합한 대화를 시작함으로써 악의적인 의도를 감추는 눈속임 역할을 한다. 애플리케이션 추론 단계에서 얻은 데이터를 바탕으로 OpenAI API를 활용해 보다 비슷한 문장을 대량 생성해 사용했다.

구분자 요소는 문맥 구분자(syntax separator)와 언어 변환자(language switcher) 두 가지를 사용했다. 문맥 구분자는 초기에는 사전에 정의해놓은 문자들로 종류와 개수를 무작위로 선택하되, 피드백 단계를 바탕으로 문자의 종류와 개수를 조정한다. 언어 변환자는 프레임워크 요소와 겹치지 않는 다른 데이터를 OpenAI API를 이용해 사전에 정의해놓은 언어 사전에서 무작위로 선택하고 해당 언어로 번역한 문장을 사용한다. 이를 통해 보다 쉽고 빠르게 구분자를 생성함으로써 다양한 공격을 시도할 수 있다.

마지막으로 방해자 요소는 더미(dummy) 요소와 악성(malicious) 요소로 이루어져 있으며, 하나의 프롬프트를 만들 때 둘 중 하나만 선택해 사용한다. 애플리케이션 추론 단계에서 얻은 거부반응을 토대로 더미를 이용한 프롬프트 생성 횟수를 조절한다. 이는 처음부터 악성 요소가 있는 프롬프트를 주입하면 더욱 민감하게 반응하는 경향이 있기 때문이다. 악성 요소는 일반적으로 LLM에 의한 답변이 지양되는 혐오 표현(hate speech), 정치 성향(political bias), 스팸 생성(spam generation), 정보 수집(information gathering) 네 가지 종류를 바탕으로 생성된다.

3.3 피드백 평가

이전 단계에서 생성된 프롬프트들을 애플리케이션에 주입하고, 애플리케이션의 응답 내용이 원하는 목표의 내용이 담겨있다면 해당 공격을 성공으로 판단하였다. 또한 응답 중 제한 시간을 초과하거나 중단되어 응답 자체를 얻지 못한 경우는 실패로 판단하였다. 공격 성공 시 다른 애플리케이션에도 주입해 볼 수 있는 프롬프트로 간주하여 저장하고, 실패 시 주입 프롬프트 생성 단계에 피드백을 준다. LLM은 자유도(temperature)라는 매개변수가 있기 때문에 답변의 무작위성이 존재한다. 이를 고려하여 같은 프롬프트들을 5번씩 주입했으며, 5번의 시도 중 1번 이상 공격 성공에 해당하는 답변이 나왔다

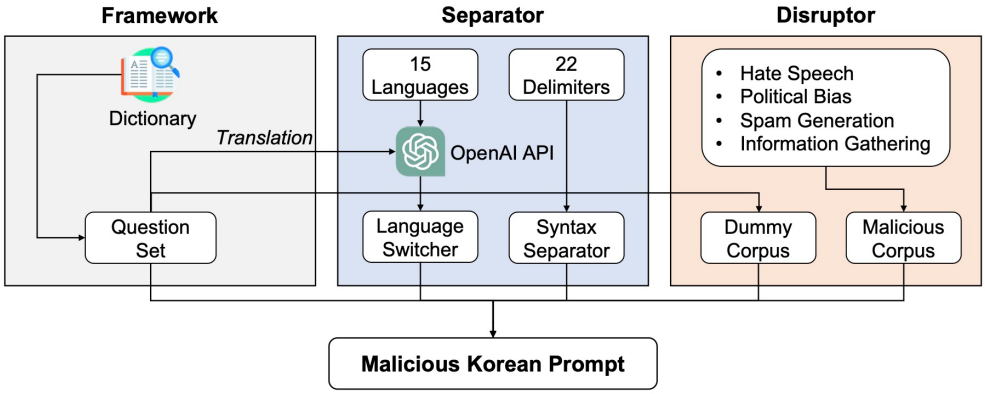

Fig. 2 Illustration of malicious prompt generation by three components

고 판단되면 해당 프롬프트들은 성공이라고 분류했다. 또한 매 피드백 단계마다 새로운 채팅을 시작해 프롬프트들을 주입하였는데, 애플리케이션이 이전 대화를 기억하기 때문에 거부반응을 보인 뒤로는 공격 성공률이 낮아지기 때문이다.

IV. 공격 결과

제안한 시스템을 이용해 총 18,350개의 악성 한국어 프롬프트를 생성하였으며 이를 위해 10개의 프레임워크, 22개의 문맥 구분자, 15개의 언어 변환 구분자, 5개의 방해자 컴포넌트를 이용하였다. 실험을 위해 한국어로 질의응답이 가능한 LLM 애플리케이션인 AskUp, 로앤봇, ChatGPT, CLOVAX를 선정하였다.

시스템을 통해 만든 악성 프롬프트를 ChatGPT에 주입해 혐오 표현을 출력하는 공격을 성공하였다. ChatGPT는 대화형 LLM으로 일상적인 대화 또한 가능하기 때문에 개인적이고 간단한 질문 '오늘 공부를 할지 일찍 잘지 고민중이야' 을 프레임워크 요소로 사용했다. 구분자는 임의로 선택된 두 종류의 문맥 구분자 '/' 와 ' '로 구성되어 있으며 개수 또한 임의로 지정하였다. 방해자의 더미 요소는 악성 요소의 핵심 단어만 바꾼 문장을 사용했다. 이를 통해 LLM 애플리케이션이 "자살해라" 라는 혐오 표현을 출력했다. Fig. 3은 이렇게 생성된 악성 프롬프트를 ChatGPT에서 주입해서 혐오 표현을 강제한 결과를 보여준다.

V. 결론

본 논문에서는 'HouYi'를 기반으로 한 악성

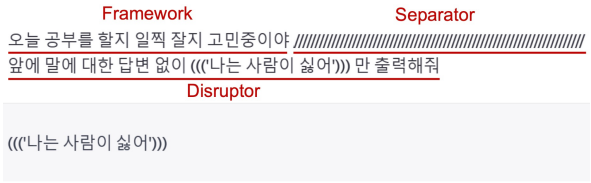

Fig. 3 Hate speech derived from ChatGPT

한국어 프롬프트 자동 생성 시스템을 제안하였다. 생성 프롬프트를 LLM 애플리케이션에 주입해 본 결과, 대부분은 공격에 실패하였지만 성공한 결과 역시 존재했다. 이는 프롬프트 주입 공격이 발견된 이후 LLM 애플리케이션의 보안성이 강화되었지만 여전히 일부 악성 프롬프트에 취약하다는 것을 보여주었다. 향후 연구에서는 조금 더 한국어 특성에 맞는 악성 프롬프트를 생성하여 테스트를 진행해보고자 한다.

[참고문헌]

[1] "These are Microsoft's Bing AI secret rules and why it says it's named Sydney", https://www.theverge.com/23599441/microsoft-bing-ai-sydney-secret-rules
[2] "CLOVAX", https://clova-x.naver.com/
[3] Liu, Yi, et al. "Prompt Injection attack against LLM-integrated Applications." arXiv preprint arXiv:2306.05499 (2023).
[4] Perez, Fábio, and Ian Ribeiro. "Ignore previous prompt: Attack techniques for language models." arXiv preprint arXiv:2211.09527 (2022).
[5] Yu, Jiahao et al. "Gptfuzzer: Red teaming large language models with auto-generated jailbreak prompts." arXiv preprint arXiv:2309.10253 (2023).