LLM을 위한 워터마크 은닉 프레임워크

김조은†, 조승현†, 김영식†, 진동섭‡

† 대구경북과학기술원, ‡ 울산대학교

jowithu@dgist.ac.kr, seunghyuncho@dgist.ac.kr, ysk@dgist.ac.kr, dsjin@ulsan.ac.kr

A Watermark Hiding Framework for Large Language Models

Jo Eun Kim†, Seung Hyun Cho†, Young-Sik Kim†, and Dongsup Jin‡

† DGIST, ‡ Ulsan University

요 약

최근 대규모 언어 모델(LLM)의 발전으로 인해 사람이 작성한 텍스트와 기계 생성 텍스트를 구별하기 어려워지며, 출처를 식별하기 위한 텍스트 워터마킹 기술이 주목받고 있다. 본 논문은 대표적인 green list 기반 워터마킹 기법인 KGW와 이를 개선한 DiPmark를 소개하고, 이들이 공통적으로 직면한 paraphrasing 공격, 탐지 정확성 저하, 텍스트 품질 손상 간의 trade-off 문제를 분석한다. 이를 해결하는 방안으로, 본 연구는 오류 정정 코드(ECC)를 활용한 워터마크 삽입 기법을 제안한다. ECC는 일부 토큰이 paraphrasing 등으로 변경되더라도 원래의 워터마크 메시지를 복원할 수 있는 잠재력을 가진다. 제안된 방식은 paraphrasing 공격에 더 강건한 워터마킹을 구현하는 데 기여할 수 있다. 향후 연구에서는 다양한 ECC 설계를 바탕으로 복원 성능과 품질 간의 균형을 실험적으로 검증할 예정이다.

Ⅰ. 서 론

최근 ChatGPT, Gemini 등과 같은 대규모 언어 모델(LLM)의 발전으로 인해, 사람이 작성한 것과 구별이 어려운 고품질의 텍스트가 자동으로 생성되고 있다[1]. 이러한 기술은 편리함을 제공하는 동시에, AI가 생성한 가짜 뉴스, 표절된 리포트, 악의적 콘텐츠 등으로 인해 다양한 사회적 문제가 제기되고 있다.

현재 LLM이 생성한 텍스트를 탐지하기 위한 다양한 방법이 연구되고 있으며, 그중에서도 텍스트 워터마크 기술이 주목받고 있다. 이 기술은 텍스트의 자연스러움을 유지하면서, 사람이 육안으로는 구별할 수 없는 미세한 통계적 또는 의미적 패턴을 텍스트에 삽입하여 생성 여부를 탐지하는 방식이다.

특히 LLM의 오·남용 가능성이 증가하는 현시점에서, 텍스트 워터마킹은 생성자 식별, 진위 검증, 저작권 보호 등 다양한 목적으로 활용 가능한 핵심기술로 부상하고 있다. 본 논문에서는 텍스트 워터마킹 기술의 개념과 기존 접근 방식을 정리하고, 이 기술이 직면한 주요 과제인 패러프레이징 공격, 탐지 정확성 저하, 텍스트 품질 손상 간의 trade-off 문제를 중점적으로 분석한다. 아울러 이러한 문제를 완화하는 방안으로, 오류 정정 부호(ECC)를 활용한 워터마크 삽입 방식을 제안하고 그 가능성을 논의한다.

Ⅱ. 기존 기술 소개 및 텍스트 워터마킹의 과제

최근 연구에서는 LLM의 어휘 집합을 두 개의 목록(green list와 red list)으로 나누고, 디코딩 과정에서 이전 토큰의 해시값에 따라 green list에 속한 토큰의 확률을 인위적으로 증가시켜 해당 토큰이 선택될 가능성을 높이는 방식의 워터마크 삽입 기법(KGW)이 제안되었다[2,3].

이 방식은 워터마크 삽입 강도와 탐지율 측면에서 효과적일 수 있으나, 다음과 같은 한계를 갖는다. 첫째, green list에 속한 토큰의 선택 비율이 높아질수록 워터마크는 강력해지지만, 동시에 공격자가 특정 토큰이 green일 가능성을 추론하기 쉬워진다. 이로 인해 공격자는 paraphrasing 등으로 워터마크를 제거하거나 우회하는 공격 전략을 설계하기 쉬워진다. 그림 1은 LLM을 이용한 paraphrasing 공격을 보여준다. 워터마크가 삽입된 텍스트가 LLM을 통해 재작성되면, 단어 순서와 표현은 변경되지만 의미는 유지되어 워터마크 탐지가 어렵게 된다.

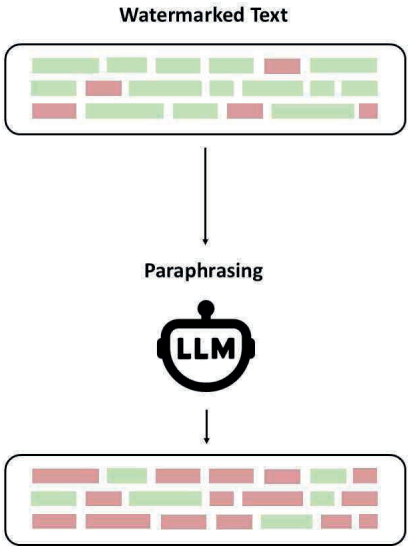

그림 1) LLM을 활용한 paraphrasing 공격

둘째, 특정 토큰의 확률을 인위적으로 조정함에 따라 출력 문장의 자연스러움이 저하될 수 있다. 예를 들어, '유 재석'이라는 이름에서 '유'가 green, '재석'이 red로 분류된다면, 디코딩 과정에서 '재석' 대신 green 리

스트에 포함된 다른 인물명(예: '지민')이 선택되어 문맥이 어색해지는 문제가 발생할 수 있다.

이러한 문제를 보완하기 위해 제안된 방식이 DiPmark이다[4]. DiPmark는 워터마크가 삽입된 분포와 원래 언어 모델의 분포 사이의 차이를 줄이기 위해, DiP-reweight을 적용한다. 이를 통해 워터마크 삽입 시에도 문장 품질을 유지할 수 있도록 한다. 그러나 red 토큰도 일정 확률로 유지되기 때문에, 워터마크의 탐지 신호는 약화되며 삽입 강도는 상대적으로 낮아지는 trade-off가 발생한다.

결국 KGW나 DiPmark 모두 paraphrasing 공격을 완전히 방어하지는 못하며, 텍스트 품질과 워터마킹 강도 사이의 trade-off 문제는 여전히 해결되지 않은 과제로 남아 있다.

Ⅲ. ECC를 활용한 워터마크 연구

ECC(Error Correcting Code)는 비트 단위의 메시지 삽입 및 복원 과정에서 오류를 탐지하고 수정할 수 있는 기능을 제공한다. 본 논문에서는 이 개념을 텍스트 워터마킹에 접목하여, 일부 단어가 paraphrasing 등으로 변경되더라도 전체 워터마크 시그니처가 복원 가능하도록 하는 방식을 제안한다.

앞서 언급한 paraphrasing 공격(그림 1)은 워터마크 패턴을 무력화 할 수 있으며, 기존 green list 기반 워터마크 방식의 주요 취약점 중 하나로 지적되어 왔다. 워터마크로 삽입할 메시지를 ECC로 인코딩하고 디코딩하면 디코딩 시 paraphrasing으로 일부 비트가 손실되더라도 오류수정 한계 내에서는 원래 메시지로 복원할 수 있다.

이러한 방식은 기존의 green list 기반 워터마크 방식이 paraphrasing 공격에 매우 취약하다는 점을 보완할 수 있으며, 워터마크에 대한 복원력이 증가하는 장점을 가지게 된다.

Ⅳ. 결론

대규모 언어 모델(LLM)의 급속한 확산으로 인해, 사람이 작성한 텍스트와 기계가 생성한 텍스트를 구별하는 문제가 점점 더 중요해지고 있다. 이에 따라, 텍스트 생성의 출처를 식별하고 오·남용을 방지하기 위한 기술로서 텍스트 워터마킹이 주목받고 있다. 본 논문에서는 현재 널리 사용되고 있는 green list 기반 워터마크 기법(KGW)과 이를 개선한 DiPmark에 대해 살펴보고, 이들이 공통적으로 직면한 주요 과제인 paraphrasing 공격 취약성, 탐지 정확도 저하, 텍스트 품질 손상 간의 trade-off 문제를 분석하였다.

이러한 한계를 극복하는 방안으로, 본 논문은 오류 정정 코드(ECC)를 활용한 워터마크 삽입 기법을 제안하였다. ECC는 삽입된 워터마크 일부가 손상되더라도 원래의 메시지를 복원할 수 있는 이점을 가지며, 이는 paraphrasing 공격과 같은 변형에 대해 보다 강건한 워터마크 복원을 가능하게 할 것으로 기대된다.

향후 연구에서는 실제 LLM 환경에서 다양한 ECC 설계와 paraphrasing 강도에 따른 복원율을 정량적으로 분석하고, 삽입할 수 있는 비트 수와 품질 보존 간의 trade-off를 실험적으로 검증할 계획이다.

ACKNOWLEDGMENT

이 논문은 2024년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (RS-2024-00442085, 자율주행 차량 서비스 보호를 위한 V2X 무선통신 인프라 보안 핵심기술 개발).

참 고 문 헌

[1] Krishna, Kalpesh, et al. "Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense." Advances in Neural Information Processing Systems 36 (2023): 27469-27500.

[2] Kirchenbauer, John, et al. "A watermark for large language models." International Conference on Machine Learning. PMLR, 2023.

[3] Kirchenbauer, John, et al. "On the reliability of watermarks for large language models." arXiv preprint arXiv:2306.04634 (2023).

[4] Wu, Yihan, et al. "A resilient and accessible distribution-preserving watermark for large language models." arXiv preprint arXiv:2310.07710 (2023).