인공지능 생성 텍스트 워터마킹 기법 분석

권기욱*, 김민석*, 구형준**
성균관대학교 (대학원생*, 교수**)

Analysis of Watermarking for AI-generated Text

Giuk Kwon*, Minseok Kim*, Hyungjoon Koo**
Sungkyunkwan University (Graduate student*, Professor**)

요 약

대규모 언어 모델 (LLM)의 발전으로 인공지능 (AI) 생성 텍스트와 인간의 글을 구별하기 어려워짐에 따라, 인공지능이 생성한 텍스트 식별과 추적 (traceability)이 가능한 워터마킹 기술의 필요성이 증가하고 있다. 본 논문은 LLM 텍스트 워터마킹의 기술 현황을 체계적으로 분석해 효과적인 워터마킹이 만족해야 할 8가지 주요 속성과 7가지 가용한 공격 방식을 정의한다. 특히 워터마크의 존재 유무를 판별하는 Zero-bit 워터마킹 (SynthID-Text, KGW, BiMarker) 기법과 복원 가능한 비트열을 텍스트에 삽입해 생성 주체를 추적하는 Multi-bit 워터마킹 (Qu et al., Cohen et al.) 기법을 알아보고, 향후 연구 방향을 제시한다.

I. 서론

대규모 언어 모델 (Large Language Model, LLM)의 발전은 인공지능 (AI)이 생성하는 텍스트를 인간의 글과 구별하기 어려운 수준에 이르게 했다 [9-10]. 특히 최신의 상용 언어 모델은 프로그래밍 코드 생성부터 학술 논문 작성, 소설 집필까지 다양한 형태의 고품질 텍스트를 생성할 수 있어 많은 사람들이 일상적 업무 전반에 활용하고 있다. 하지만 이러한 기술 발전은 학술 부정행위, 가짜 뉴스 생성, 저작권 침해 등과 같은 사회적 문제 또한 야기할 수 있으며, 이에 따라 인공지능이 생성한 글을 식별하고 추적할 수 있는 텍스트 워터마킹 기술의 필요성이 증가하고 있다.

텍스트 워터마킹은 이미지나 오디오와 다른 고유한 도전 과제를 안고 있다. 첫째, 텍스트는 토큰 단위로 구성된 이산적 (discrete) 데이터이므로, 토큰 하나만 바뀌어도 문장의 의미가 크게 달라진다. 둘째, 언어의 낮은 엔트로피 특성으로 인해 삽입 가능한 정보량이 제한적이다. 셋째, 패러프레이징이나 번역 같은 의미 보존 변환에 대한 강건성을 확보하기 어렵다.

본 논문은 이러한 LLM 텍스트 워터마킹의 최신 동향을 체계적으로 조사해, 8가지 핵심 속성과 7가지 공격 방식을 제시하고 최신 워터마크 기법들의 설계 원리를 살펴본다.

II. AI 생성 텍스트 워터마킹의 속성과 공격 방식

본 장에서는 기존의 서베이 (Survey) 논문들 [5-7]의 분석을 바탕으로, 워터마킹 기법이 갖추어야 할 핵심 속성 8가지와 주요 공격 방식 7가지를 체계적으로 정의한다.

2.1 핵심 속성

강건성 (Robustness): 단어의 삭제 및 추가, 동의어 치환, 패러프레이징 등 다양한 텍스트

본 논문은 2025년도 정부(과학기술정보통신부) 한국연구재단 (생성 모델 컴플라이언스를 위한 모듈형 AI 워터마킹 기술 연구실; No.RS-2025-02293072) 지원으로 수행한 연구임.

변형 공격 이후에도 워터마크를 검출할 수 있는 능력을 의미한다. ***텍스트 품질 (Text quality):*** 워터마크 삽입 후에도 AI 생성 텍스트의 자연스러움, 유창성, 의미적 일관성이 유지되는 정도를 나타낸다. ***삽입 정보 용량 (Payload Capacity):*** 워터마크 존재 여부를 넘어 사용자 ID, 모델 버전 등의 메타데이터를 텍스트에 삽입할 수 있는 비트 수를 의미한다. ***오픈소스 적용성 (Open-source Applicability):*** 다양한 오픈소스 모델들의 서로 다른 아키텍처, 배포 방식에도 범용적으로 적용 가능한 호환성과 이식성을 나타낸다. ***공개 검증 가능성 (Public Verifiability):*** 워터마크 삽입 및 탐지 과정의 투명성과 제3자에 의한 검증 가능성을 의미한다. ***실용성 (Practical Deployment):*** 실시간 텍스트 생성 환경의 계산 및 메모리 오버헤드 측면에서 워터마킹 모델을 현실적으로 적용할 수 있는 능력을 나타낸다. ***엔트로피 적응성 (Entropy Adaptivity):*** 텍스트의 엔트로피 수준에 따라 워터마크 삽입 및 탐지를 유연하게 조정할 수 있는 능력을 의미한다. ***신뢰할 수 있는 검출력 (Reliable Detectability):*** 워터마크 탐지 시 낮은 오탐율 (false positive/negative rate)과 높은 일치율 (match rate)을 달성하는 능력을 나타낸다.

이러한 8가지 핵심 속성은 워터마킹 기술의 신뢰성과 실용성을 결정하는 중요한 요소이며 각 속성은 서로 상충관계를 가지고 있다. 따라서 응용 시나리오에 맞추어 우선순위를 정하고 적절한 균형점을 찾아가는 것이 중요하다.

2.2 공격 방식

LLM 텍스트 워터마킹 시스템의 안정성을 평가하기 위해서는 다음과 같은 7가지 공격 시나리오를 고려해야 한다. ***삭제 공격 (Deletion Attack):*** 워터마크된 텍스트에서 선택적으로 단어 또는 문장을 제거해서 워터마크 신호를 약화시키거나 파괴하는 공격이다. ***추가 공격 (Addition Attack):*** 새로운 단어 또는 문장을 텍스트에 삽입해 워터마크 패턴을 교란시키고, 워터마크된 토큰의 비율을 탐지 임계값 이하로 낮추는 전략이다. ***치환 공격 (Replacement Attack):*** 워터마크된 토큰들을 의미적으로 가까운 동의어나 유사 표현으로 교체하는 공격이다. ***재구성 공격 (Rephrase Attack):*** 텍스트 전체의 의미를 유지하면서 완전히 다른 구조와 어휘로 문장을 재작성하는 공격으로, 패러프레이징 모델이나 다른 LLM을 활용할 수 있다. ***동형 문자 공격 (Homoglyph Attack):*** 시각적으로 유사하지만 유니코드상 다른 문자로 치환하는 공격이다. (예: 라틴 문자 'o'를 키릴 문자 'о'로 교체). ***비밀키 추출 공격 (Secret Extraction):*** 다수의 워터마크된 입·출력 쌍의 패턴 분석 등을 통해 워터마킹 시스템의 비밀키를 역공학적으로 추출하는 공격이다. 성공 시 워터마크 위조와 제거가 모두 가능해진다. ***스푸핑 공격 (Spoofing Attack):*** 워터마크된 텍스트의 대량 수집을 통해 대리 모델 (surrogate model)을 학습시켜 워터마크 패턴을 모방하는 공격이다. 이를 활용해 악의적인 콘텐츠를 특정 모델이 생성한 것처럼 위장 가능하다.

평가 기준	세부 항목
핵심 속성	강건성, 텍스트 품질, 삽입 정보 용량, 오픈소스 적용성, 공개 검증 가능성, 실용성, 엔트로피 적응성, 신뢰할 수 있는 검출력
공격 방식	단어 및 문장 삭제·추가·치환, 텍스트 재구성, 동형 문자 치환, 비밀키 추출, 스푸핑

표 1. AI 생성 텍스트 워터마킹의 속성과 공격 방식

위와 같은 7가지 공격은 단독으로 또는 여러 가지가 조합되어 사용될 수 있으며, 각 공격의 효과는 워터마킹 알고리즘의 구현 방식과 파라미터 설정에 따라 달라진다. 따라서 신뢰할 수 있는 워터마킹 시스템을 설계하기 위해서는 표 1에서의 핵심 속성과 공격 방식을 종합적으로 고려하는 것이 필수적이다.

III. 주요 워터마킹 기법

3.1 제로비트 워터마킹

제로비트 워터마킹은 워터마크의 존재 여부만을 탐지하는 기법으로, 추가적인 메타데이터 없이 텍스트가 특정 모델에 의해 생성되었는지만 판별한다. 이러한 제로비트 방식은 LLM의

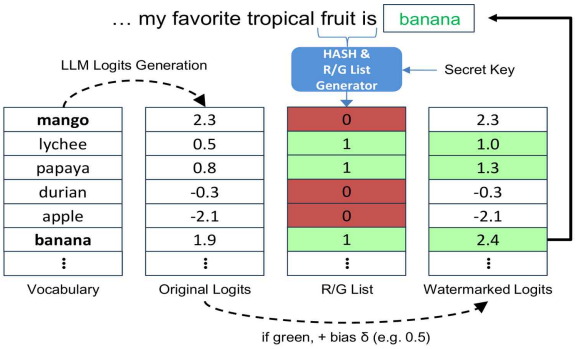

그림 1. KGW [8] 워터마킹 알고리즘. LLM이 다음 토큰의 로짓 (logit) 값을 계산하는 단계에서 이전 토큰과 비밀키를 입력으로 갖는 해시 함수를 활용해 vocabulary를 녹색/적색 영역으로 나누고, 녹색으로 분류된 항목의 로짓에 편향을 주어 워터마크를 삽입한다.

토큰 생성 과정 중 어느 단계에서 삽입되는지에 따라 크게 토큰 샘플링 단계와 로짓 (logit) 생성 단계로 분류할 수 있다. 본 절에서는 실제 프로덕션 환경에 적용된 SynthID-Text [3], 녹색/적색 리스트를 통해 로짓 편향을 주는 워터마크 체계인 KGW [8], 그리고 KGW의 통계적 검출 정확도를 개선한 BiMarker [4] 등 대표적인 제로비트 기반 기법들을 살펴본다.

토큰 샘플링 단계. 모델이 다음 토큰 후보들의 로짓 값을 계산한 후, 최종 토큰을 선택하는 샘플링 (sampling) 단계에서 워터마크를 삽입하는 방식이다. Dathathri et al. [3]이 제안한 SynthID-Text는 토너먼트 (tournament) 샘플링 알고리즘을 통해 워터마크를 삽입한다. 토너먼트 샘플링은 LLM의 확률 분포에서 2^m개의 후보 토큰을 추출한 후, m개 레이어에 걸쳐 토너먼트를 진행하며 각 레이어는 서로 다른 입력의 의사 난수 함수를 사용해 토큰 별 승자를 선택한다. 이 과정을 통해 텍스트 품질 저하 없이 워터마크를 자연스럽게 삽입할 수 있으며, 이는 실제 대규모 프로덕션 환경인 Gemini에 적용된 생성형 텍스트 워터마킹 시스템이다.

로짓 생성 단계. 모델이 다음 토큰으로 올 단어들의 로짓을 계산하는 단계에서 워터마크를 삽입하는 방식이다. Kirchenbauer et al. [8]의 KGW 기법은 그림 1에서 나타내는 것과 같이 각 토큰 위치에서 어휘를 녹색/적색 영역으로

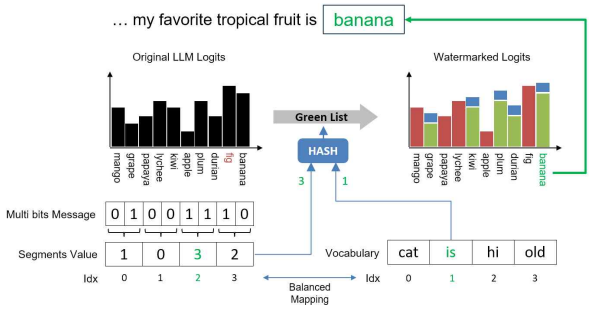

그림 2. Qu et al. [1] 워터마킹 알고리즘. 삽입하려는 메시지 정보 (segments)와 vocabulary 항목을 균등하게 대응 (mapping)한 다음, 워터마크 편향을 주기 위한 해시 함수의 입력으로 메시지 정보를 넣어준다.

동적 분할한다. 이전 토큰과 비밀키를 입력으로 갖는 해시 함수를 통해 어휘의 절반을 녹색 영역으로 지정한 후, 녹색에 해당하는 항목의 로짓 값에 편향 δ를 추가한다. 즉, 비밀키를 갖고 있는 사람만이 재현할 수 있는 편향 값을 통해 통계적으로 검출 가능한 녹색 토큰 비율의 편차가 텍스트에 내재된다. 이 방법은 단순하며 효과적이지만 낮은 엔트로피의 텍스트에서는 편향 δ의 효과가 줄어들기 때문에 녹색 토큰 비율의 편차가 감소해 검출 안정성이 떨어진다. 이를 해결하기 위해 BiMarker [4]는 생성 텍스트를 양극 (positive pole)과 음극 (negative pole)으로 나누어, 양극에서는 녹색 항목의 로짓을, 음극에서는 적색 항목의 로짓을 증가시킨다. 탐지 시에는 두 극성 간 녹색 토큰 수의 차이를 계산하여 워터마크를 판별함으로써, 낮은 강도 워터마크의 추출 부정확성을 감소시켰다.

3.2 멀티비트 워터마킹

멀티비트 워터마킹은 워터마크 존재 여부 판단뿐만 아니라 사용자 식별자, LLM 모델 버전 등의 추가 정보를 텍스트에 삽입하는 확장된 워터마킹 방식이다. 이러한 멀티비트 방식은 삽입하려는 정보 (payload)를 어떤 단위로 구성하고 텍스트 전체에 어떻게 분산시키는지에 따라, 블록 (block) 기반과 세그먼트 (segment) 기반으로 나뉜다. 본 절에서는 다중 사용자 추적과 엔트로피 적응성을 고려한 Cohen et al. [2]의 프레임워크와, 오류 정정 기법이 적용된 세그먼

트 기반 정보 삽입의 강건성을 증명한 Qu et al. [1]의 접근법을 중심으로 논의한다.

블록 기반. 생성되는 텍스트를 특정 길이의 블록 단위로 나눠 워터마크를 삽입하는 기법이다. Cohen et al. [2]은 0과 1로 이루어진 L-bits 메시지의 비트 i마다 두 개의 키 ($k_{i,0}$, $k_{i,1}$)를 만들어 총 2L개의 키를 갖고, 무작위 인덱스 i를 뽑아 해당 키로 충분한 엔트로피를 만족하는 블록을 생성·연결한다. 그 결과, 짧은 텍스트에서는 기반 zero-bit 방식의 검출 보장을 유지하며, 긴 텍스트 (블록이 충분히 많음)에서는 메시지 추출이 가능해진다. 또한 핑거프린팅 코드 (fingerprinting code) 기술을 활용해 한 사용자뿐만 아니라 공모 집단 (다중 사용자)에 대해서도 추적할 수 있는 프레임워크를 설계했다.

세그먼트 기반. 워터마크로 삽입하려는 메시지를 여러 개의 세그먼트로 분할하여 텍스트 전체에 분산시키는 방식이다. 그림 2의 Qu et al. [1]은 메시지를 여러 비트의 세그먼트로 나누고, 각 vocabulary 항목에 균등하게 분배하는 방식 (balanced mapping)을 제안한다. 이를 통해 특정 비트에 토큰이 편중되는 기존 세그먼트 할당 방법의 불균형 문제를 해소하고 검출 정확도를 증가시켰다. 또한 Reed-Solomon 오류정정 코드 (ECC)를 적용하여 텍스트 편집에 대한 강건성을 향상시켰으며, 편집 거리 (edit distance) 내에서의 워터마크 복원 가능성을 데이터 세트별로 증명함으로써, 실제 환경에서의 수학적 강건성 보장을 강화했다.

IV. 논의 및 향후 연구 방향

텍스트 워터마킹 기술은 워터마크 존재 여부를 판별하는 제로비트에서 워터마크 생성 주체의 추적까지 가능한 멀티비트로 발전했으며, 신뢰할 수 있는 검출력, 텍스트 품질, 강건성, 실용성, 엔트로피 적응성의 속성 측면에서 문제를 해결하는 데에 기여했다. 하지만 공개 검증 가능성, 오픈소스 적용성의 측면에서는 여전히 충분한 고려가 이루어지지 않았으며, 다양한 속성과 공격 방식 전반에 대해 정량적으로 평가된 워터마킹 방식이 부재했다. 따라서 앞으로의 연구는 간과된 속성을 만족시키는 범용 워터마킹 프레임워크의 설계와 핵심 속성 및 공격 방식 전반에 대한 공정한 평가 체계 구축에 초점을 맞추어야 한다.

V. 결론

본 논문은 LLM 텍스트 워터마킹의 핵심 속성 8가지와 주요 공격 방식 7가지를 체계적으로 정리하며, 대표적인 제로비트 및 멀티비트 워터마킹 기법들의 설계 원리와 특징을 분석했다. 이 연구가 텍스트 워터마킹의 평가 기준 체계화와 모델 개선을 위한 기초 자료로 활용되기를 기대한다.

[참고문헌]

[1] Wenjie Qu et al., Provably Robust Multi-bit Watermarking for AI-generated Text, USENIX, 2025.

[2] Aloni Cohen et al., Watermarking Language Models for Many Adaptive Users, IEEE S&P, 2025.

[3] Sumanth Dathathri et al., Scalable watermarking for identifying large language model outputs, Nature, 2024.

[4] Zhuang Li et al., BiMarker: Enhancing Text Watermark Detection for Large Language Models with Bipolar Watermarks, arXiv, 2025.

[5] Yuqing Liang et al., Watermarking Techniques for Large Language Models: A Survey, arXiv, 2024.

[6] Aiwei Liu et al., A Survey of Text Watermarking in the Era of Large Language Models, ACM Computing Surveys, 2024.

[7] Xuandong Zhao et al., SoK: Watermarking for AI-Generated Content, IEEE S&P, 2025.

[8] John Kirchenbauer et al., A Watermark for Large Language Models, ICML, 2023.

[9] Maurice Jakesch et al., Human heuristics for AI-generated language are flawed, Proceedings of the National Academy of Sciences, 2023.

[10] Elizabeth Clark et al., All That's 'Human' Is Not Gold: Evaluating Human Evaluation of Generated Text, ACL IJCNLP, 2021.