# ChatGPT와 Wikipedia 간 악성 코드 정보 문서 유사도 분석*

송승호[1†], 최지영[1†], 김진우[2‡]
[1,2]광운대학교 (학부생, 교수)

# Analyzing the Document Similarity of Malware Information between ChatGPT and Wikipedia

Seung-Ho Song[1†], Ji-Young Choi[1†], Jin-Woo Kim[2‡]
[1,2]Kwangwoon University(Undergraduate Student, Professor)

**요 약**

최근 생성형 거대 언어 모델(Large Language Model, LLM)의 성능이 비약적으로 발전하여 다양한 분야에서 이용되고 있다. 그러나 대부분 LLM은 사실이 아닌 정보를 제공하는 '할루시네이션'이라는 문제를 가지고 있는데, 이는 정보의 정확성을 요구하는 보안 도메인에서 치명적이다. 본 논문에서는 오늘날 가장 대중적으로 이용되고 있는 LLM 애플리케이션인 ChatGPT가 제공하는 악성 코드 정보의 정확성을 검증하고자 한다. 이를 위해 294개의 악성 코드 정보를 ChatGPT와 Wkipedia에 질의하고 두 문서의 유사도 분석을 수행하였다. 분석 결과 ChatGPT는 악성 코드 정보의 절반 이상이 Wikipedia의 정보와 70%의 유사도를 나타내는 것을 볼 수 있었다.

## I. 서론

최근 몇 년 동안 생성형 거대 언어 모델(Large Language Model, LLM)은 텍스트 생성 및 이해 분야에서 비약적인 성능 발전을 이루었다. 2032년에는 LLM과 같은 생성형 인공지능 모델의 시장 규모가 약 1조 3천억 달러로 성장할 것으로 예측되기도 하였다[1].

그러나 한편으로 LLM은 때로 거짓 정보를 생성하거나 본래와는 다른 정보를 제공할 수 있는데 이를 '할루시네이션(hallucination)'이라고 한다[2]. 할루시네이션은 정보의 정확성이 중요한 보안 분야에서는 더 심각한 문제로 여겨질 수 있다. 특히 최근에는 LLM을 OSINT (Open Source Intelligence)에 적용하여 빠르고 효율적으로 보안에 관련된 정보를 수집하여 위협을 탐지하는 방안이 고려되고 있다[3]. 만약 OSINT와 같이 수집한 정보의 신뢰도가 중요한 분야에서 LLM이 잘못된 보안 정보를 제공한다면, 위협에 대한 대책을 잘못 수립할 수 있다. 따라서 LLM의 보안 도메인에 대한 할루시네이션 정도를 사전에 분석하고 평가하는 것이 중요하다.

본 논문에서는 현재 가장 많이 이용되는 LLM 애플리케이션인 ChatGPT를 대상으로 '악성 코드'에 관한 정보를 질의하고 이를 Wikipedia에 있는 정보와 유사도를 비교하였다. 실험을 위해 자연어 처리에서 활용되는 유사도 비교 방법을 활용하였다. 이를 통해 ChatGPT가 보안 도메인에 관한 정보를 신뢰성 있게 제공하는지를 알 수 있을 것으로 기대한다.

* 이 논문은 2023년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. RS-2022-00166401)
† 공동 1저자
‡ 교신저자, jinwookim@kw.ac.kr

## II. 배경 지식 및 관련 연구

### 2.1 Cyber Threat Intelligence (CTI)

CTI는 사이버 위협에 관한 광범위한 증거를 수집하고 처리하여 정교한 사이버 공격을 탐지하

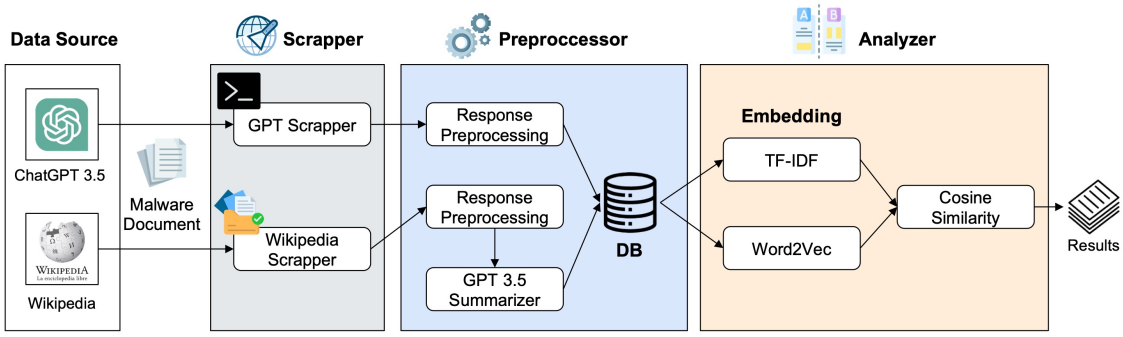


Fig. 1 The proposed system that consists of a scrapper, a preprocessor, and an analyzer

거나 그에 대한 대책을 마련하는 포괄적인 방법론을 의미한다[4].

OSINT는 SNS, 블로그, 기사 등 '공개된 출처'로부터 정보를 수집하는 CTI의 한 분야이다. 최근 ChatGPT와 같은 LLM 애플리케이션을 OSINT에 활용하려는 시도가 제안되기도 하였다[3, 5, 6]. 이는 LLM이 기존에 사람이 하던 것보다 더 빠르고 정확하게 정보를 수집하고 가공할 수 있기 때문이다. 특히 LLM의 대표적인 유스케이스는 텍스트 요약(text summarization), 질의응답(question answering)을 들 수 있는데 이들은 OSINT에서 사람이 악성 코드에 대한 정보를 수집할 때 하는 작업과 유사하다.

### 2.2 정보 격차와 할루시네이션

OSINT는 이른바 '정보 격차(information gap)'라는 문제를 가지고 있는데[8], 이는 정보의 출처가 다른 내용의 데이터를 제공할 때 발생한다. 예를 들어 2011년, 악성 코드 Shylock[9]의 발견 시기를 출처 A는 2월, 출처 B는 9월이라고 주장하는 경우를 들 수 있다[7]. 한편, 이러한 문제는 LLM을 활용한 OSINT에서도 '할루시네이션' 현상으로 유사하게 발생할 수 있다. 이는 모델이 거짓 정보를 사실인 양 생성하는 현상을 지칭한다[2]. 이를 보안 도메인에 적용한다면 LLM이 악성 코드 정보를 사실과 다르게 가공하여 제공할 수 있음을 의미한다. 만약 보안 전문가가 LLM 모델로부터 악성 코드에 관한 잘못된 정보를 제공받는다면, 해당 위협에 대해 잘못된 보안 대책을 수립할 수 있게 된다. 따라서 LLM 애플리케이션의 보안 도메인에 관한 정보의 정확성을 사전에 분석하고 평가하는 것이 중요하다.

## III. 분석 방법

본 논문에서는 대표적인 LLM 애플리케이션인 ChatGPT를 대상으로 악성 코드 정보에 대한 정확성을 분석하였다. 이를 위해 Wikipedia가 제공하는 정보를 사실(ground-truth)로 가정하고, 이를 ChatGPT가 생성한 정보와 텍스트 유사도를 비교하는 시스템을 설계하였다. Wikipedia에 작성된 정보들은 이미 많은 연구를 통해 일정 수준 이상의 신뢰성이 검증되었다[11]. 다만 본 연구는 ChatGPT가 생성하는 정보가 Wikipedia와 같이 일반적으로 사용되는 지식 기반(knowledge base)의 정보와 유사한지를 평가하는데에 의의를 둔다. 따라서, Wikipedia에 기재된 정보들의 사실 여부를 완전히 보장할 수 없을지라도 본 연구에 활용되기에 충분하다고 판단하였다. Fig. 1은 텍스트 유사도 분석을 위한 시스템 설계도이며 각각 1) 수집기(scrapper), 2) 전처리기(preprocessor), 3) 분석기(analyzer)로 구성된다.

### 3.1 악성 코드 문서 수집

먼저 Wikipedia와 ChatGPT에 악성 코드 정보를 수집하였다. 동일한 정보들에 대해 질의하기 위해 먼저 Wikipedia가 악성 코드 정보에 대해 제공하는 기사(article) 중에서 총 294개를 선별하고 이를 '악성 코드 집합'으로 정의하였다.

이후 집합에서 각 악성 코드명을 순차적으로 Wikipedia로부터 관련 문서를 수집하였다. 기본적으로 해당 Wikipedia 페이지 내의 본문 내용을 전부 수집했다. 그러나 본문 내용이 많을 시

에는 해당 악성 코드에 대한 핵심 내용들을 희석할 수 있다. 이에 Wikipedia의 특성상 단어에 대한 요약 및 정의에 대한 내용이 문서 상단에 위치한다는 점을 이용하여, 일정 길이 이상의 내용들은 상위 세 개의 문단만 수집하였다. 이와 동시에 동일한 악성 코드 정보를 ChatGPT에 질의하였다. 이때 OpenAI가 제공하는 GPT-3.5-turbo 모델의 API[10]를 사용하였으며, 모델의 파라미터를 적절하게 조절함으로써 응답의 일관성을 유지했다.

ChatGPT에 질의 시, 답변 길이를 적절하게 제한하게끔 프롬프트(prompt)를 작성했다. OSINT에 LLM 애플리케이션을 적용할 때는 악성 코드 정보에 대해 짧고 간결한 정보(예: 발견 일시, 침투 방법 등)를 파악하길 원할 것으로 가정하였다. 이외에는 별다른 제한을 두지 않음으로써 모델의 답변 자유도를 보장하였다.

### 3.2 응답 문서 전처리

앞에서 질의해서 얻은 응답 문서는 유사도 분석을 위해 정형화된 형식으로 변환이 필요하다. 공통적으로 자연어 처리 전 수행하는 소문자로 변환, 불용어와 구두점 제거, 어근 추출(lemmatization) 과정을 수행한다. 정형화된 문서들은 모두 이후 유사도 분석을 위해 데이터베이스에 저장하였다.

이 때 Wikipedia에 질의해서 얻은 응답 문서를 GPT 3.5 모델을 이용해 요약하는 과정을 거치도록 하였다. 일반적으로 Wikipedia의 글은 상세하고 방대한 정보를 담고 있어서, 그대로 사용하기에는 길고 복잡하다. 따라서 요약하지 않았을 경우 불필요한 정보 때문에 유사도 분석 결과가 왜곡될 수 있다. 또한 ChatGPT의 응답 문서와 유사하게 길이를 맞춤으로써 유사도 분석 시간과 저장 공간을 줄일 수도 있다.

이 때 요약하는 과정에서 오류를 최대한 배제하기 위해 다음과 같은 프롬프트 엔지니어링(prompt engineering)을 수행하였다. 첫째, 오직 문서 내에 주어진 단어들만을 가지고 요약하도록 하고 GPT 모델의 고유 응답이나 의견은 배제하도록 하였다. 둘째, 원본 데이터는 Wikipedia 사이트에서 가져왔음을 명시하였다.

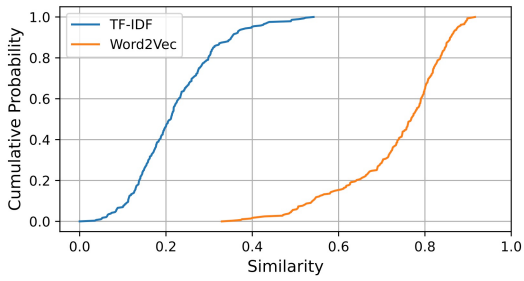


Fig. 2 The cumulative distribution function (CDF) for cosine similarity

셋째, 요약문의 길이를 조정하여(예: 30 단어) GPT에서 얻은 응답 길이와 유사하게끔 했다.

### 3.3 문서 간 유사도 분석

마지막으로 수집한 데이터 간의 텍스트를 비교하여 유사도를 분석하였다. 이를 위해 자연어 처리에서 흔히 문서 간 유사도를 비교할 때 사용되는 코사인 유사도(cosine similarity) 기법을 사용하였다. 코사인 유사도는 벡터화한 문서 간의 유사도를 계산하는 방법이다. 즉, $n$ 차원의 문서 벡터 $A$, $B$에 대한 코사인 유사도는 다음과 같이 정의된다.

$$S(A,B)=\cos(\theta)=\frac{A\cdot B}{\|A\|\|B\|}=\frac{\sum_{i}^{n}A_i\times B_i}{\sqrt{\sum_{i}^{n}(A_i)^2}\times\sqrt{\sum_{i}^{n}(B_i)^2}}$$

이때 $A_i$, $B_i$는 각 벡터의 $i$번째 원소를 의미한다.

문서를 벡터화하는 임베딩(embedding) 방법에 따라 유사도 결과가 다르게 나타날 수 있다. 본 논문에서는 널리 사용되는 임베딩 방법인 TF-IDF (Term Frequency Inverse Document Frequency)와 Word2Vec 모델을 적용하였다. TF-IDF는 각 단어가 문서 내에서 얼마나 중요한지를 측정하는 통계적 방법이다. 이를 활용하여 전처리된 텍스트 문서를 TF-IDF 벡터로 변환한 후 두 벡터 간의 코사인 유사도를 측정하였다. TF-IDF는 단순히 단어 빈도만을 비교하기 때문에 Wikipedia와 ChatGPT가 제공한 악성 코드 정보의 길이가 다를 경우 유사성이 낮게 측정될 수 있다. 이를 보완하기 위한 임베딩 방법으로 Word2Vec을 사용하였다. Word2Vec은 단어를 고차원 벡터 공간에 매핑하는 데 사용되는 기계 학습 모델로, 문맥상 유사한 단어가

벡터 공간에서 서로 가깝게 위치하도록 하는 것이 특징이다. 각 문서를 Word2Vec을 이용하여 문장 내의 단어 벡터의 평균으로 표현 후, 두 벡터 간 코사인 유사도를 계산하였다.

Fig. 2는 294개의 악성 코드에 대해 질의한 결과를 TF-IDF와 Word2Vec으로 임베딩 한 후 코사인 유사도를 산출한 것을 나타낸 CDF이다. 결과적으로 악성 코드 정보 중 절반 이상이 Word2Vec 방법에서 70% 이상의 유사도를 보이는 결과를 볼 수 있다. ChatGPT가 제공하는 악성 코드 정보는 Wikipedia의 정보와 어느 정도 유사함을 알 수 있다. 또한, TF-IDF 방식으로 임베딩 한 결과는 Word2Vec 방식보다 낮은 유사도를 보임을 관찰할 수 있었다. TF-IDF 방식은 Bag-of-Words 모델을 기반으로 하며, 문서 내에서 각 단어가 나타나는 빈도수를 통해 중요도를 계산한다. 이 방식은 문장의 구조나 단어 간의 유기적인 관계를 고려하지 않은 정량적 방법이기 때문에, 문장의 의미를 고려하지 않고 계산한 결과가 나왔다.

## IV. 결론

본 논문에서는 ChatGPT와 Wikipedia 간 악성 코드 정보 유사성을 자연어 처리 기법을 통해 분석하였다. 실험 결과, ChatGPT가 제공한 악성 코드 정보가 Wikipedia와 일정한 유사도를 보임을 확인할 수 있었다. 특히 Word2Vec 방식을 사용한 임베딩은 TF-IDF 방식에 비해 더 높은 유사성이 나타났으며, 이는 단어 간 의미적 유사성을 고려할 때 ChatGPT의 악성 코드 정보가 어느 정도 신뢰성이 높다는 점을 보여준다.

다만 본 연구는 다양한 LLM 애플리케이션 및 다른 보안 정보 출처와의 비교에 대한 연구가 미흡하다. 또한, 실험에 사용된 악성 코드 정보의 범위와 양이 제한적일 수 있으며, 다양한 보안 도메인에서의 결과를 보장하지는 못한다. 또한 본 연구에서는 코사인 유사도를 사용하여 악성 코드 정보의 유사성을 분석하였으나, 통계적 기법의 근본적인 한계로 문맥상 유사성을 정확하게 판단하지 못한다는 한계점이 있다. 이를 개선하기 위해서는 모델 학습을 통해 문맥적으로 유사도 판정 성능을 향상시켜야 한다.

마지막으로 본 연구가 LLM 애플리케이션의 정보 신뢰성을 평가하는 초석이 될 것으로 기대한다.

## [참고문헌]

[1] “Generative AI to Become a $1.3 Trillion Market by 2032, Research Finds”, https://www.bloomberg.com/company/press/generative-ai-to-become-a-1-3-trillion-market-by-2032-research-finds/

[2] Ji, Ziwei, et al. "Survey of hallucination in natural language generation." ACM Computing Surveys 55.12 (2023): 1-38.

[3] Singla, Tanmay, et al. "An Empirical Study on Using Large Language Models to Analyze Software Supply Chain Security Failures." arXiv preprint arXiv:2308.04898 (2023).

[4] Tounsi, Wiem, and Helmi Rais. "A survey on technical threat intelligence in the age of sophisticated cyber attacks." Computers & security 72 (2018): 212-233.

[5] “Using the Power of ChatGPT for OSINT”, https://blog.sociallinks.io/using-the-power-of-chatgpt-for-osint/

[6] “Automating the Tasks of OSINT Analysts with Chatbot and ChatGPT: A Deep Dive”, https://www.linkedin.com/pulse/automating-tasks-osint-analysts-chatbot-chatgpt-deep-dive-groeneveld/

[7] Jo, Hyeonseong, et al. "GapFinder: Finding inconsistency of security information from unstructured text." IEEE Transactions on Information Forensics and Security 16 (2020): 86-99.

[8] Jiang, Yuning, Manfred Jeusfeld, and Jianguo Ding. "Evaluating the data inconsistency of open-source vulnerability repositories." Proceedings of the 16th International Conference on Availability, Reliability and Security. 2021.

[9] “Shylock Banking Trojan”, https://www.kaspersky.com/resource-center/threats/shylock-banking-trojan-definition

[10] “OpenAI API Reference”, https://platform.openai.com/docs/api-reference

[11] Aniket Kittur, et al. “Can You Ever Trust a Wiki? Impacting Perceived Trustworthiness in Wikipedia” ACM Computer supported cooperative work (2008): 477-480