참고 Toxicity Knowledge CoT 개발

□ 연구 개요

○ 배경

- 신약 후보 발굴을 위해 새로운 화학적 구조와 약리학적 특성을 가진 화합물의 지속적 탐색이 필요
- 기존 화합물의 성능 한계를 극복하고, 난치성 질환 치료 및 새로운 치료제 개발을 위해 새로운 화학 구조발굴이 필수적임
- 그러나 기존의 독성 평가 방식은 실험 기반으로 진행되기 때문에 시간·비용이 많이 들고, 초기 스크리닝 단계에서 대규모 후보군을 평가하는 데 한계를 지님. 또한 고전적인 머신러닝 모델이나 예측기반 QSAR 모델은 예측값만 제공할 뿐, 왜 특정 분자 구조가 독성을 유발하는지에 대한 설명(설명가능성, XAI)이 부족하여 규제 대응과 신뢰성 확보가 어려운 상황 임.
- 이러한 배경에서 화학 분자 구조 정보를 기반으로 독성 여부를 예측하고, 그 판단 근거를 자연어로 설명할 수 있는 AI 기반 독성 추론 모델의 필요성이 커지고 있음.
- 특히 최근의 LLM 기술 발전은 단순한 예측을 넘어 추론(reasoning) 의 수행을 가능하게 함. 그 중에서도 Chain-of-Thought(CoT) 형식은 모델이 판단 절차를 단계별로 서술하도록 유도하여, 독성의 원인·기전·구조적 특징을 보다 투명하게 드러내는 데 효과적인 기술임.
- 이러한 이유로, 분자 구조 기반의 독성 추론 데이터를 활용한 LLM 모델은 단순한 예측 도구를 넘어 설명 가능한 형태로 발전가능성을 지님.

○ 관련 연구

- CoTox (2025): CoTox는 분자 독성 예측에 LLM 기반 Chain-of-Thought(CoT) reasoning을 도입한 모델로, 화학 구조 정보와 생물학적 지식을 함께 활용하여 독성 여부와 단계적 추론 과정을 동시에 생성하는 모델임.
- An Explainable Supervised Machine Learning Model for Predicting Respiratory Toxicity of Chemicals Using Optimal Molecular Descriptors (2022) : 이 모델은 분자 구조에서 추출한 descriptor 기반 특징을 사용해 호흡기 독성 여부를 예측하는 supervised ML 모델임. 또한 SHAP과 같은 XAI 기법을 활용해 독성 예측에 기여한 중요 특징(feature importance) 을 제시함으로써 설명 가능성을 제공함.

○ 기존 연구의 한계점 및 개선 사항

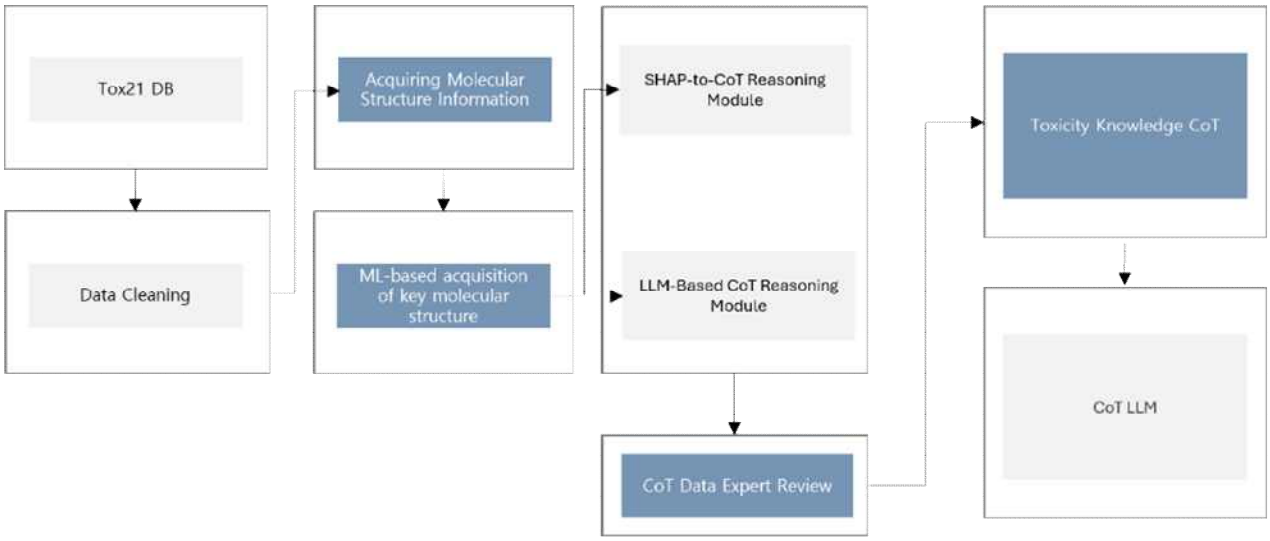
- 기존 CoT 기반 독성 예측 모델은 LLM이 직접 Chain-of-Thought를 생성함으로써 분자 독성에 대한 추론을 가능하게 했지만, 이 reasoning이 구조적 근거(feature evidence)에 기반하지 못한다는 한계를 지님. 즉, 독성 유발 기전과 연결된 명확한 증거 없이 언어 모델이 서술을 생성하기 때문에 기전적 타당성과 신뢰성이 부족함. 또한 이를 검증하는 전문가 검수 과정도 포함되지 않음.
- SHAP 기반의 설명성을 도입한 연구들도 있으나, 설명이 수치적 feature importance 수준에 머물러 있음. 독성의 원인·구조적 기전 등을 자연어 reasoning으로 해석하는 기능은 제공하지 않음. 또한 descriptor 기반 예측 모델이므로 구조적 다양성과 기전적 설명에 한계를 가짐.

○ 연구 목표

- 본 연구에서는 **근거 기반 고신뢰 독성 정보 추론 (Toxicity Knowledge CoT Reasoning) 데이터를** 구축하기 위해 다음과 같은 통합 프레임워크를 제안하였음.
  (1) SHAP 기반 구조적 근거(evidence)를 추출. 변수 간 상호작용과 독성 유발 요인을 정량적으로 확보
  (2) 도출된 SHAP 근거를 **rule-based reasoning 형태로 구조화**하여 기전적 추론 단위를 정의
  (3) 정형화된 규칙 기반 근거를 LLM에 입력하여, **자연어 Chain-of-Thought(CoT)**으로 확장·서술하도록 하는 생성 구조를 설계
  (4) 생성된 reasoning은 **전문가 검수(expert-in-the-loop)**과정을 통해 생물학적 타당성과 기전적 정확성을 보정
- 본 프레임워크를 적용하여 생성된 데이터는 기존 모델들이 제공하지 못한 **근거 기반 (evidence-grounded)·고신뢰(high-reliability) 독성 추론 체계**를 구현하는 것을 목표로 함.
- 이를 토대로 독성 예측 LLM 모델을 추가적으로 구현하여 성능을 확인 하고자 함.

□ 주요 설계

○ Architecture 구성도



○ 수도 코드 (Pesudo code)

```
input  : toxicity_input.json
output : toxicity_output.json (model_response + LABEL + accuracy)

#1. Load Input Toxicity Data
data ← load_json(input_path)

#2. Batch Inference via FastAPI-vLLM Server
model_outputs ← run_batched(data)
```

```
#3. Attach Model Responses + Ground-Truth Labels
for each (sample, out):
    sample.model_response ← out
    sample.LABEL ← 1 if sample.activity == "Active" else 0

#4. Evaluate Prediction Accuracy
accuracy ← compute_accuracy(model_response, LABEL)

#5. Save Output JSON
save_json(data, output_path)
return accuracy
```

□ 기술 개발 결과

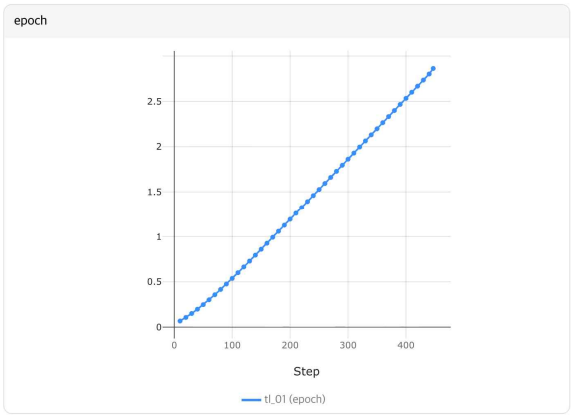
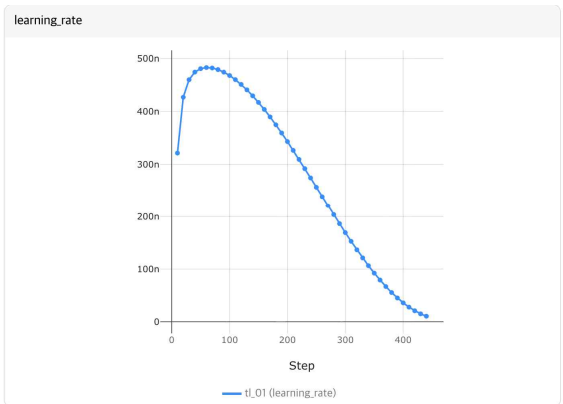
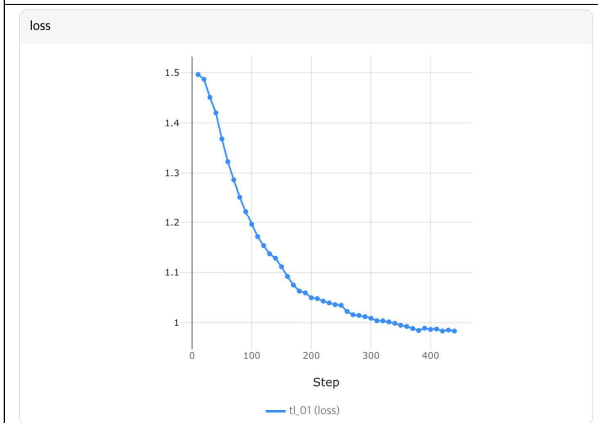
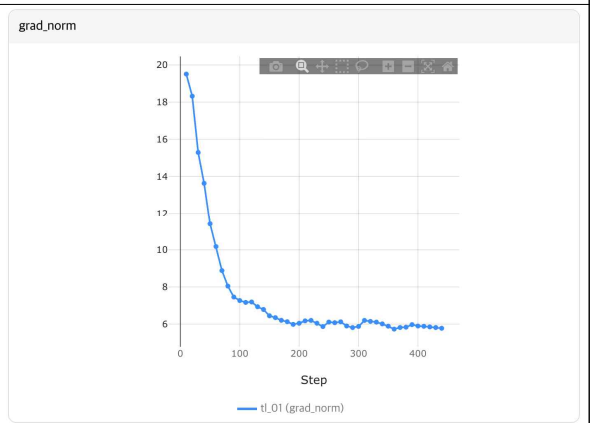
○ 주요 성능 지표(Performance Metrics)

- 본 연구에서 개발한 고신뢰 독성 정보 추론(Toxicity Knowledge CoT Reasoning) 기반 분자 독성 예측 모델은 대규모 독성 데이터셋을 활용하여 학습을 수행하였음., FastAPI-vLLM 기반의 배치 추론 프레임워크를 적용하여 모델의 예측 성능을 정량적으로 평가하였음.
- 모델은 독성 활성(Active/Inactive)에 대한 이진 분류 과제를 수행.
- 모델 성능 요약

· Accuracy : 90.9% ± 0.5

· 이는 기존 RNN/Transformer 기반 독성 예측 모델 대비 높은 일관성과 신뢰도를 확보한 성능으로, SHAP 기반 근거(evidence)와 기전적 추론(mechanistic reasoning)을 결합한 프레임워크의 효과가 확인됨.

-

• epoch	• learning_rate
	
• loss	• grad_norm
	

○ 소스코드 및 주석

```
# main.py
import os
import sys
import shlex
import time
import argparse
import random
from datetime import datetime
from utils import load_json, save_list_to_json, FileLogger
from evaluate import run_batched
from scoring import evaluate_accuracy_only
def main():
    parser = argparse.ArgumentParser(description="Batch inference runner (requests) for custom FastAPI vLLM server")
    parser.add_argument("--input", default=os.getenv("INPUT_JSON", "/path/to/input.json"))
    parser.add_argument("--output", default=os.getenv("OUTPUT_JSON", "/path/to/output.json"))
```

```
    parser.add_argument("--log_dir", default=os.getenv("LOG_DIR", "./logs"))
    parser.add_argument("--retries", type=int, default=int(os.getenv("RETRIES", "2")))
    parser.add_argument("--batch_size", type=int, default=int(os.getenv("BATCH_SIZE", "32")))
    parser.add_argument("--timeout", type=float, default=float(os.getenv("REQ_TIMEOUT", "300")))
    args = parser.parse_args()
    base_url = os.getenv("INFER_BASE_URL", "http://localhost:30001")
    infer_batch_path = os.getenv("INFER_BATCH_PATH", "/generate_batch")
    run_id = datetime.now().strftime("%Y%m%d_%H%M%S")
    log_path = os.path.join(args.log_dir, f"run_{run_id}.txt")
    logger = FileLogger(log_path)
    cmdline = " ".join(shlex.quote(x) for x in sys.argv)
    logger.log(f"Command: {cmdline}")
    logger.log(f"Input: {args.input}")
    logger.log(f"Output: {args.output}")
    logger.log(f"Base URL: {base_url}")
    logger.log(f"Batch path: {infer_batch_path}")
    logger.log(f"Batch size: {args.batch_size}, Retries: {args.retries}, Timeout: {args.timeout}")
    random.seed(42)
    t0 = time.time()
    data = load_json(args.input)
    logger.log(f"Loaded {len(data)} records from input.")
    results = run_batched(
        data=data,
        base_url=base_url,
        path=infer_batch_path,
        batch_size=args.batch_size,
        logger=logger,
        retries=args.retries,
        request_timeout=args.timeout,
    )
    for d, out in zip(data, results):
        d["model_response"] = out
        d["LABEL"] = 1 if d["compound_info"]["toxicity"]["activity"] == "Active" else 0
    acc = evaluate_accuracy_only(
        data, answer_key="model_response", label_key="LABEL", count_missing_as_wrong=False
    )
    logger.log(f"Accuracy (scored only): {acc:.4f}")
    save_list_to_json(data, args.output)
    dt = time.time() - t0
    logger.log(f"Saved results to {args.output}")
    logger.log(f"Elapsed: {dt:.2f} sec")
```

```
    logger.log("Done.")
if __name__ == "__main__":
    main()
```

○ Git-Hub 공개

- https://github.com/KwangSun-Ryu/ADMET-AGI-Toxicity-Knowledge-COT--