NetConfigQA: 네트워크 설정 해석 능력 평가를 위한 질의응답 데이터셋

박유진[1]*, 진경은[1]*, 박찬진[2], 김기현[2], 김태훈[1], 전윤호[1], 박천음[1]†
[1]국립한밭대학교, [2]한국과학기술정보연구원
{20191906, 20227005}@edu.hanbat.ac.kr
{thkim, yhjeon, parkce}@hanbat.ac.kr

NetConfigQA: A Question-Answering Dataset for Network Configuration Interpretation

Yujin Park[1]*, Gyeongeun Jin[1]*, Chanjin Park[2], Kihyeon Kim[2], Taehoon Kim[1], Yunho Jeon[1], Cheoneum Park[1]†
[1]Hanbat National University, [2]Korea Institute of Science and Technology Information

요 약

본 연구는 운영 중 네트워크 설정의 역방향 해석을 지원하기 위해 NetConfigQA 데이터셋과 평가 파이프라인을 제안한다. PnetLab XML을 표준화하여 12개 카테고리 75개 필수 문항 기반의 기초 질문과 페르소나 심화 질문을 포함한 총 813문항을 구성하였으며, 정답은 스크립트와 LLM기반의 정답 생성기를 통해 정답을 생성하고 근거 경로를 기록하여 직접 검증하였다. GPT-4o-mini를 대상으로 Baseline, Chain of Thought(CoT), 본 논문에서 제안하는 파이프라인 세 가지 접근 방식을 비교한 결과, 제안하는 방식이 EM 0.819, F1 0.837로 가장 우수한 성능을 보였다.

주제어: LLM, 네트워크 매니지먼트, 네트워크 설정 해석, 토폴로지 특화 질의응답, PnetLab, NetConfigQA, RAT

1. 서론

네트워크 관리는 안정적이고 효율적인 운영을 위해 필수적이나, 대규모 환경에서는 장비와 프로토콜의 다양성, 복잡한 구성으로 관리 부담이 가중된다. 기존 스크립트 기반 접근 방식은 동적 변화에 취약하며, AI-ML 기법 역시 비정형 데이터 처리 한계로 최적화된 의사결정을 보장하지 못한다. 최근 부상한 거대언어모델(LLM)은 네트워크 설정 해석, 질의응답, 오류 탐지 등에서 높은 활용 잠재력을 지니며, 운영 자동화와 이상 감지, 예측 유지보수에 기여할 수 있는 잠재력을 가진다.

그러나 기존 연구는 주로 "자연어 요구사항 → 설정 생성"에 집중해 왔다. 예를 들어 NetConfEval [1]은 자연어 명령을 Cisco IOS 설정으로 변환하고, NeMoEval [2]은 토폴로지 분석 코드를 생성하는 등 대부분이 구성을 '만드는' 데 초점이 맞춰져 있다. 최근 조사 연구에 따르면 LLM의 네트워킹 응용이 Intent-based Networking(IBN)과 같은 자동화 시나리오에 집중되어 있음을 지적한다 [3]. 하지만 실제 운영에서는 이미 배포된 수많은 설정을 해석하고 검증하는 해석 중심 접근 방식이 더 중요하다. 예를 들어, "BGP 설정의 단일 장애점은 어디인가?", "VRF 불일치가 서비스에 미치는 영향은 무엇인가?"와 같은 질문에 답할 수 있어야 한다. 조사 논문 역시 생성 중심 접근법의 한계와 토폴로지 해석, 검증 영역의 부족을 지적하며 [3] 상태 질의와 설정 검증을 위한 데이터셋 필요성을 강조한다.

이러한 배경 속에서 본 연구는 PnetLab[4] 환경의 로그파일(네트워크 장치 설정)을 입력 받아 해석 중심 질의응답을 수행하는 NetConfigQA 데이터셋과 평가 파이프라인을 제안한다.

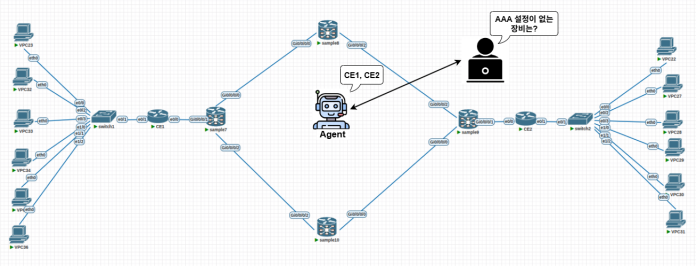

그림 1. LLM을 활용한 네트워크 상태 확인 과정

이는 LLM의 네트워크 관리 도메인 이해 능력을 검증하고, 해석 중심 네트워크 관리를 위한 토대를 제공한다.

2. 관련연구

최근 대규모 언어 모델(LLM)을 네트워크 관리와 구성 자동화에 적용하려는 연구가 활발히 진행되고 있다 [3]. 이러한 연구들은 크게 LLM이 수행하는 핵심 작업을 기준으로 (1) 새로운 설정을 만들어내는 생성 중심 접근법과, (2) 기존 설정을 분석하고 이해하는 해석 중심 접근법으로 나눌 수 있다.

[표 1]은 본 연구와 선행 연구의 평가 초점을 요약한다. 선행 연구 다수는 자연어 요구사항을 입력으로 받아 설정이나 코드를 생성하는 시나리오에 집중되어 있다 [1, 5, 6, 2]. 그러나 이러한 접근은 운영 환경에 배포된 기존 설정을 대상으로 한 상세 해석, 정책적 타당성 검증, 잠재적 위험 추론 수행에 한계를 가진다. 본 논문은 이러한 문제를 해결하기 위해 네트워크 상태에 대한 해석 중심 방법을 제안한다. LLM의 네트워크 관리 도메인 이해 능력을 평가하기 위해 질의응답 데이터셋 NetConfigQA를 구축한다. NetConfigQA는 PnetLab [4] 로그 파일(네트워크

*Equal contribution
†Corresponding author

표 1. LLM 기반 네트워크 관리 벤치마크 비교

벤치마크	주요 목표	입출력	데이터 생성	평가 방식	토폴로지 특화
NetConfEval [1]	설정 · 코드 생성	요구→설정/코드/API	반자동	정답 일치	X
NETPRESS [5]	에이전트 운영	지시→액션	자동	정확 · 안전 · 지연	△
NETLLMBENCH [6]	명령 생성	지시→JSON(단일)	수동	에뮬레이터 검증	△
NeMoEval [2]	토폴로지 분석코드	질의→그래프코드	수동	실행결과 정확	X
NetConfigQA (Ours)	설정 해석 · QA	네트워크 설정→질문, 답	반자동	정답 일치	**O**

설정 XML 파일)을 입력으로 받아 네트워크 망의 상태, 정책, 설정값 등에 관한 질문과 정답을 포함한다.

3. 연구방법

본 논문은 LLM의 네트워크 관리 도메인 이해 능력을 평가하기 위해, 네트워크 장치 설정 파일을 기반으로 질의응답 데이터셋을 자동 생성하고 이를 이용한 평가 파이프라인을 제시한다. 먼저 네트워크 장비 설정에서 핵심 정보를 추출하고, 각 장치별 서식 차이를 공통 JSON 구조로 정규화한다. 이후 정규화된 정보를 바탕으로 질문과 정답을 자동 생성하여 평가 데이터셋을 구축하며 구축한 데이터셋을 활용한 질의응답 평가 파이프라인을 통해 LLM의 네트워크 설정 해석 능력을 정량적으로 평가한다.

3.1 질문 생성 파이프라인

3.1.1 XML 표준화

질문 생성 파이프라인을 다루기 앞서 Cisco IOS, IOS-XR, NSO 등 각 플랫폼별 XML 서식 차이를 해소하기 위해 XML 서식을 공통 JSON 구조로 표준화하며, 예시는 [표 2]에 제시한다.

표 2. 플랫폼별 JSON 표준화 예시

IOS
XML: `<Ethernet><name>0/0.100</name></Ethernet>`
JSON: `interfaces.name="Ethernet0/0.100"`

IOS-XR
XML: `<GigabitEthernet><id>0/0/0/1.100</id></GigabitEthernet>`
JSON: `interfaces.name="GigabitEthernet0/0/0/1.100"`

이를 통해 플랫폼 독립적 표현을 확립하고, 추가 변환 없이 후속 단계가 원활히 수행되도록 하는 기반을 마련한다.

3.1.2 기초 질문 생성

기초 질문 생성 절차는 세 가지로 구분된다. (1) [표 3]와 같이 12개 카테고리, 75개 추출 정보를 정의한다(예: BGP AS 번호, VRF 개수, SSH 활성화, 인터페이스 구성 등). (2) 정의된 각 항목을 기반으로 공통 질문 서식을 스크립트로 생성하며, 장치, VRF, AS 등 대상 엔티티를 입력으로 받아 문장 구조를 자동 완성하도록 설계한다[표 4]. (3) 실제 네트워크 장치의 메타데이터를 서식에 주입해 질문 인스턴스를 대량 생성한다. 이러한

표 3. 주요 카테고리별 추출 정보

카테고리	추출 정보 수	추출 정보 예시
Security Policy	5	SSH 활성화 장비
BGP Consistency	4	AS iBGP 이웃 장비
VRF Consistency	4	VRF RD 값
L2VPN Consistency	4	L2VPN 페어
OSPF Consistency	3	OSPF 인터페이스
System Inventory	5	장비 호스트네임
Security Inventory	4	SSH 활성화 여부
Interface Inventory	5	장비 인터페이스 개수
Routing Inventory	4	로컬 BGP AS 번호
Services Inventory	7	VRF 이름
Basic Info	14	시스템 호스트네임
Command Generation	17	BGP 요약 확인 명령어
총합	**75**	

표 4. 서식 별 기초 질문 예시

서식	질문 예시
`SSH` 활성화 장비 관련	장비 1번에 SSH가 활성화되었는가?
`iBGP` 누락 관련	AS 2번의 iBGP 누락 피어는?
`eBGP` 장비 관련	1번 장비 eBGP 연결 IP 주소?.
`OSBF` 프로세스 관련	1번 장비 OSPF 프로세스 ID 목록?.
장치별 `VRF` 관련	1번 장비에 설정된 VRF는 총 몇 개?
`OSPF` 인터페이스 관련	1번 장비에 연결된 인터페이스는?

과정으로 생성된 기초 질문은 추론이나 외부 문맥을 요구하지 않고 설정에서 직접 확인 가능한 정보만을 다루므로 정답 근거가 명확하며 장치가 달라져도 문장 구조의 일관성을 유지 할 수 있다.

3.1.3 심화 질문 생성

심화 질문 생성 단계는 추론 심도와 평가 신뢰성 확보를 위해 두 가지로 구분된다. (1)에서는 앞선 단계와 동일하게 네트워크 설정으로부터 필요한 정보를 추출한다. (2)에서는 LLM에 5개 역할(네트워크 엔지니어, 문제 해결 전문가, NOC 운영자, 보안 감사자, 아키텍트)을 부여하고, 페르소나 기반 프롬프트를 통해 각 역할별 심화 질문을 생성한다. 이때 입력은 추출된 설정 정보와 장비 메타데이터로 구성되며, 출력은 각 역할 관점의

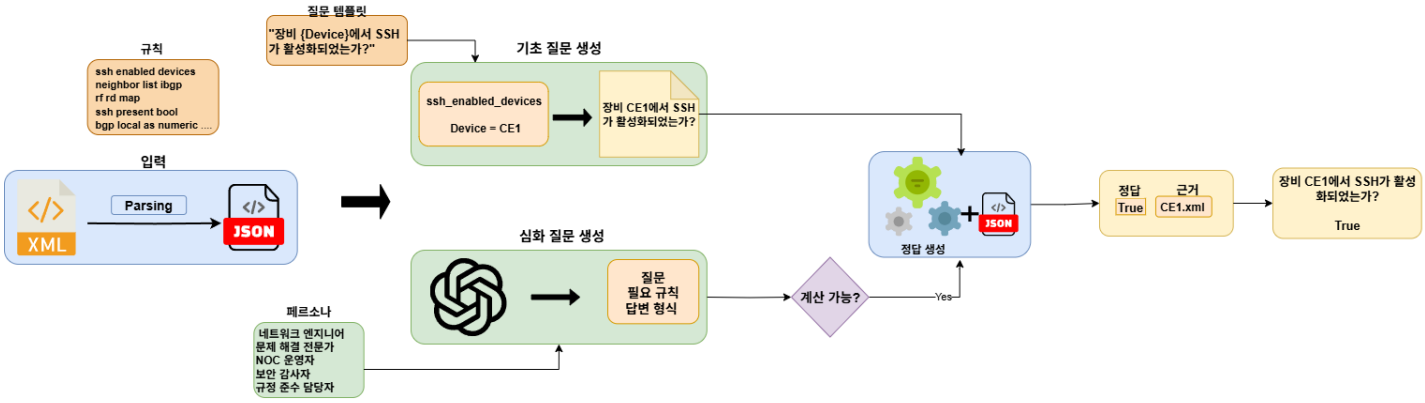

그림 2. 기초 질문 생성 과정

심화 질문, 참조 장비의 메타데이터, 기대 정답 형식을 포함한다. LLM으로 인한 왜곡을 최소화 하기 위해 모든 생성은 추출 정보에 기반하여 수행한다. 예시는 [표 5] 와 같다.

표 5. 심화 질문 예시

질문 예시
CE1–sample7 링크가 다운될 때, sample7 → sample9의 최단 경로와 경로 변화 여부는?
현재 구성에서 sample7의 CE-향 서브인터페이스 Gi0/0/0/1.100의 VRF 바인딩과 경로 연동을 확인하기 위한 CLI를 올바른 순서로 제시하세요.
모든 PE 라우터(sample7/8/9/10)의 VRF exam-l3vpn에 Import/Export RT 65000:1000이 모두 설정되어 있습니까?.
sample10–sample7 코어 링크가 다운될 때, sample10 → sample8 도달이 가능합니까?

3.1.4 정답 생성 및 검토

대량으로 생성된 기초 질문의 정답은 스크립트 기반으로 생성된다. 스크립트 기반으로 생성된 질문에는 추출 정보, 대상 장비가 포함되어 있다. 이를 통해 스크립트 기반의 정답 계산기를 호출하여 네트워크 설정에서 직접 정답을 계산한다. 최종적으로 질문, 정답, 대상 장비를 함께 저장해 (1) 질문, (2) 정답, (3) 장비로 구성된다.

심화 질문은 기초 질문과 달리 LLM 기반 정답 생성 과정이 적용된다. LLM은 질문에 포함된 추출 정보를 바탕으로 정답 생성에 필요한 정보를 스크립트 기반 계산 엔진으로 계산하여 각 장비의 네트워크 설정에서 증거를 수집한다. 수집된 증거를 바탕으로 LLM이 최종 정답과 간단한 설명을 생성한다. 산출되는 출력은 (1) 정답, (2) 설명, (3) 장비로 구성된다.

모든 답변은 네트워크 설정에서 재현 가능한 계산 절차로 도출되어야 한다는 제약을 적용하며, 외부 상식이나 모델 추론만으로 채워지는 항목은 허용하지 않는다.

이후 검토 단계에서 수기 점검을 수행한다. 각 문항에 대해 질문, 정답, 설명을 확인하여 오류, 모호성이 발견되면 문항을 개선하거나 폐기한다.

표 6. NetConfigQA 데이터셋 구성

구분	질문 수	비율	평균 토큰	총 토큰
기초 질문	763	93.8%	30.8	23521
심화 질문	50	6.2%	51.8	2589
총계	813	100%	32.1	26110

3.2 LLM 질의응답 파이프라인

본 논문에서는 네트워크 장비 설정 파일 및 관련 문서를 활용하여 다양한 유형의 질의를 처리할 수 있는 LLM 기반 질의응답 파이프라인을 제안한다. 제안하는 파이프라인은 단순 조회와 기타 복합 과제로 분류하여 각각의 처리 전략을 다르게 적용한다. 전체 구조는 [그림 3]과 같이 (1) 작업 분류, (2) 반복적 답변 개선, (3) 최종 응답 최적화 총 세 단계로 나뉜다.

반복적 답변 개선 단계에서는 초안 답변 T_n을 기반으로 관련 증거 집합 V_n과 참조 문맥 I_n을 동적으로 검색 후 응답을 보정한다. 이를 통해 RAT(Retrieval-Augmented Thoughts) [7] 구조를 반영하여 각 반복에서 핵심 결론의 정확도와 서술의 일관성을 단계적으로 향상시킨다.

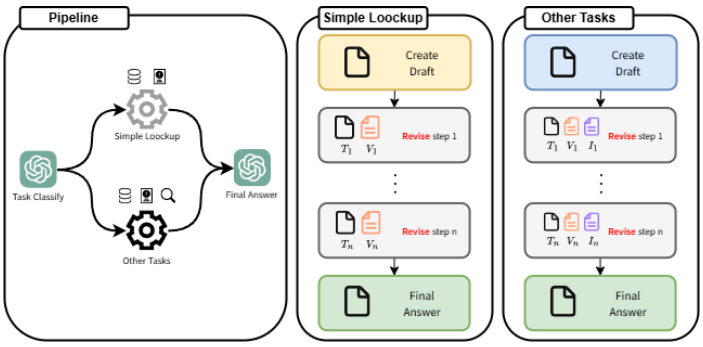

그림 3. 질의 응답 파이프라인

3.2.1 작업 분류

분류 단계의 목적은 입력 질의를 단순 조회와 기타 복합 과제로 이진 분류하여 파이프라인의 라우팅을 결정하는 작업이다.

본 논문은 다음과 같은 분류 범주에 따라 판정 원칙을 규정하고 이를 바탕으로 효과적인 과제 분류 체계를 구축한다.

분류 범주

- **단순 조회**: 네트워크 장비 설정으로부터 직접 조회로 충분한 경우. (예: 장비 상태, 라우팅 테이블, 인터페이스 등)
- **기타 복합 과제**: 네트워크 장비 설정 참조만으로 충분하지 않은 경우. (예: 설정 변경, 정책 적용, 장애 분석, 원인 파악 및 복구, 구성 및 정책 검토, 구조 최적화, 보안 준수 점검 등)

3.2.2 반복적 답변 개선

기존 CoT의 경우 단순 조회 질의에서는 높은 일치도를 보이나, 복합 과제의 경우 근거 부족 혹은 표현 부족 현상이 나타남을 확인하였다. 이에 본 논문은 초안 생성 후 참조 문서와 XML 구성을 통해 검증된 사실을 단계적으로 주입하여 초안으로 부터 정정하는 반복적 답변 개선 절차를 적용한다. 이를 통해 매 반복에서 핵심 결론의 정확도와 서술의 일관성을 단계적으로 향상시킨다.

3.2.3 최종 응답 최적화

단순 조회의 경우 요구되는 값을 간결하게 제시함으로써 빠르고 명확한 정답 확인이 가능하도록 하였으며, 복합 과제의 경우 핵심 결론을 첫 줄에 간결히 제시하고 그 뒤에 근거, 설정 방법, 절차, 보안 고려 사항과 모범 사례를 체계화하고 근거 정보를 함께 제공함으로써 사용자가 결과를 검증 수 있도록 구성한다.

출력 원칙

- **단순 조회**: 질의가 요구하는 값이나 장비명을 불필요한 설명 없이 단일 값 혹은 콤마 구분 목록으로만 출력한다.
- **기타 복합 과제**: 정답 부분은 핵심 결론을 간결히 담고, 기술적 세부 사항에는 설정 명령, 절차, 보안 고려 사항, 모범 사례 및 근거 정보를 포함한다.

4. 실험

본 논문에서는 NetConfigQA 데이터셋을 활용하여 LLM의 네트워크 설정 해석 능력을 체계적으로 평가한다. 데이터셋은 PnetLab [4]에서 수집한 실제 네트워크 장비 XML 로그 파일을 기반으로 총 813개의 질의응답 쌍으로 구성되며, 기초 질문 763개(93.8%)와 심화 질문 50개(6.2%)로 분류된다. 평가 모델은 GPT-4O-mini 모델을 사용하여 진행한다.

4.1 방법에 따른 성능 평가 결과

실험을 통해 Baseline, CoT, 그리고 제안하는 방식이 평가 모델의 정답률에 미치는 영향을 정량적으로 비교한다. BERTScore는 설명이 포함된 심화 질문에 한해 산출한다.

표 7. 방법론별 성능 비교 결과

평가 지표	Baseline	CoT	Ours
EM	0.666	0.688	**0.819**
F1	0.707	0.733	**0.837**
BERTScore	0.878	**0.879**	0.819

[표 7]을 보면,제안 방법(Ours)이 **EM 0.819**, **F1 0.837**로 가장 높다. Baseline 대비 EM 0.153, F1 0.130, CoT 대비 각각 EM 0.131, F1 0.104 개선을 보여준다. 반면 BERTScore에서는 CoT가 0.879로 가장 높은 점수를 기록했으며, 이는 CoT가 의미적 유사성에서 높은반면, 정확도(EM/F1)에서는 제안 방식이 더 강하다는 점을 보여준다.

4.2 정답률과 설명의 질적 상관관계 분석

이 실험을 통해 EM/F1만으로는 답변의 옳고 그름의 이유가 드러나지 않는 한계를 보완하기 위해, 설명 품질(BERTScore)과 정확도 사이의 상관관계를 점검한다. [표 7]을 기준으로 Baseline과 CoT의 응답에 대해 BERTScore를 산출해 상, 중, 하 3분위로 나누고 각 분위별로 EM/F1을 비교하여 설명 품질이 높을수록 정답 일치도(EM)와 부분 일치도(F1)가 함께 향상하는지 평가한다.

표 8. 설명 품질(BERTScore)별 정확도 비교

방법	BERTScore 상위 그룹	BERTScore 중위 그룹	BERTScore 하위 그룹
	EM / F1	EM / F1	EM / F1
Baseline	0.438 / 0.550	0.312 / 0.660	0.294 / 0.498
CoT	**0.500** / 0.542	0.125 / 0.447	0.188 / 0.605
Ours	0.250 / 0.461	0.222 / **0.658**	0.100 / 0.561

[표 8]는 설명 품질과 정답률 간의 상관관계를 보인다. 방법별로 보면, EM은 대체로 BERTScore 상위 그룹이 가장 높게 나타난다. 반면 F1은 상 · 중 · 하에서 BERTScore가 증가할수록 증가하지 않고 중위 혹은 하위 그룹에서 더 높게 관찰되기도 한다. 이는 F1이 부분 일치를 평가하는 특성상, 핵심 개념은 포함하지만 세부 계산이나 구체적 결과가 틀린 중위 그룹에서 높은 점수를 받을 수 있기 때문이다. 그럼에도 전반적으로 BERTScore가 높을수록 EM의 정답 일치도가 함께 개선되는 경향은 일관되게 나타난다.

[표 9]는 BERTScore 상/중/하 대표 사례를 보여준다. 상위 그룹은 링크 단절 시 경로 단절이라는 정보를 근거와 함께 정확히 제시하며 정답도 일치한다. 반면 중위/하위 그룹은 설명 문장 자체의 타당성은 있으나, 실제 토폴로지/이웃 매핑/조합 계산이 틀려 정답이 어긋난다.

표 9. 정성 분석 예시

그룹(BERTScore)	질문(요약)	GT	예측	정확	설명 요약 (GT / 예측)
상위	sample9–sample8 코어 링크가 다운될 때, sample9 → CE1 도달이 가능한가?	불가능	불가능	O	**GT:** 정상 경로는 sample9→sample8→sample7→CE1. sample9의 코어 이웃은 사실상 sample8뿐이므로 해당 링크 다운 시 CE1로 우회 불가. **예측:** sample9–sample8 링크 단절 시 CE1 경로 차단(동일 근거).
중위	AS 65000의 iBGP 풀메시 구성은 완전합니까? (정답 형식: TRUE/FALSE)	TRUE	FALSE	X	**GT:** 4대 PE가 서로 3개 이웃씩 맺어 조합 6쌍 모두 존재 ⇒ 풀메시 완전. **예측:** 일부 장비만 연결되어 있다고 가정하여 미완전 판정.
하위	AS 65000의 iBGP Full-Mesh에서 누락된 피어 쌍 수는? (정답 형식: 0 이상의 정수)	0	3	X	**GT:** 장비 4대 ⇒ 고유 쌍 6개, 설정상 전부 존재 ⇒ 누락 0. **예측:** 3쌍만 있다고 가정해 누락 3으로 오답.

5. 결론

본 논문은 네트워크 설정의 해석 중심 접근 방법을 지원하기 위해 NetConfigQA 데이터셋과 평가 파이프라인을 제안하였다. 제안한 데이터셋은 규칙 기반 기초 질문과 LLM 생성 및 규칙 기반 평가로 산출된 심화 질문을 포함하여, 재현성과 확장성을 동시에 확보하였다. 실험 결과, 정확도(EM/F1)는 제안 방법이 가장 높고, CoT는 설명의 의미적 정합성(BERTScore)에서 우수한 성능을 보였다. 이 결과는 LLM이 네트워크 관리 도메인의 설정 해석, 검증 과제에 활용될 수 있는 가능성을 보여준다. 향후 연구에서는 더 다양한 상황과 토폴로지를 포함하는 데이터셋 확장과 실제 운영 환경 적용성 검증을 진행할 예정이다.

참고문헌

[1] C. Wang, M. Scazzariello, A. Farshin, S. Ferlin, D. Kostić, and M. Chiesa, "Netconfeval: Can llms facilitate network configuration?" *Proc. ACM Netw.*, Vol. 2, No. CoNEXT2, Jun. 2024. [Online]. Available: https://doi.org/10.1145/3656296

[2] S. K. Mani, Y. Zhou, K. Hsieh, S. Segarra, T. Eberl, E. Azulai, I. Frizler, R. Chandra, and S. Kandula, "Enhancing network management using code generated by large language models," *ACM Workshop on Hot Topics in Networks (HotNets)*, November 2023.

[3] G. O. Boateng, H. Sami, A. Alagha, H. Elmekki, A. Hammoud, R. Mizouni, A. Mourad, H. Otrok, J. Bentahar, S. Muhaidat *et al.*, "A survey on large language models for communication, network, and service management: Application insights, challenges, and future directions," *IEEE Communications Surveys & Tutorials*, 2025.

[4] [Online]. Available: https://pnetlab.com/pages/documentation

[5] Y. Zhou, J. Ruan, E. S. Wang, S. Fouladi, F. Y. Yan, K. Hsieh, and Z. Liu, "Netpress: Dynamically generated llm benchmarks for network applications," *arXiv preprint arXiv:2506.03231*, 2025.

[6] K. Aykurt, A. Blenk, and W. Kellerer, "Netllmbench: A benchmark framework for large language models in network configuration tasks," *2024 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, pp. 1–6, 2024.

[7] Z. Wang, A. Liu, H. Lin, J. Li, X. Ma, and Y. Liang, "Rat: Retrieval augmented thoughts elicit context-aware reasoning in long-horizon generation," *arXiv preprint arXiv:2403.05313*, 2024.

[8] Y. Wei, X. Xie, Y. Zuo, T. Hu, X. Chen, K. Chi, and Y. Cui, "Leveraging llm agents for translating network configurations," 2025.

[9] K. Shen, Y. Zhang *et al.*, "What do llms need to synthesize correct router configurations?" *Proceedings of the ACM Workshop on Hot Topics in Networks (HotNets)*, 2023.

[10] A. Abane, A. Battou, and M. Merzouki, "An adaptable ai assistant for network management," *Proceedings of the IEEE/IFIP Network Operations and Management Symposium (NOMS)*, 2024.

[11] Y. Shen, J. Shao, X. Zhang, Z. Lin, H. Pan, D. Li, J. Zhang, and K. B. Letaief, "Large language models empowered autonomous edge ai for connected intelligence," *arXiv preprint arXiv:2307.02779*, 2023.

[12] D. Bilò, K. Choudhary, S. Cohen, T. Friedrich, and M. Schirneck, "Efficient fault-tolerant search by fast indexing of subnetworks," *arXiv preprint arXiv:2412.17776*, 2024.