

NFV 관리를 위한 LLM 기반 Intent Translation 방법 연구

홍 지 범*, 홍 원 기°

A Study of LLM-based Intent Translation Method for NFV Management

Jibum Hong* and James Won-Ki Hong°

요 약

오늘날 네트워크 규모가 커지고 복잡도가 증가하면서, 사용자의 의도 (intent)를 기반으로 네트워크를 효율적으로 관리하기 위한 Intent-based Networking (IBN) 개념이 제안되었다. 최근 대규모 언어 모델 (Large Language Model, LLM)의 발전으로 IBN에 LLM을 적용하여 자연어 intent를 시스템이 해석할 수 있는 구조화된 형태로 변환해 주는 Intent Translation 기술이 연구되고 있다. 본 논문에서는 Intent Translation을 통해 Network Function Virtualization (NFV)를 관리하기 위한 자연어 intent를 추상화 및 구조화하는 템플릿을 생성하는 방법을 제안한다. 제안하는 방법은 프롬프트 엔지니어링 기법 중 Chain-of-Thought (CoT), few-shot learning을 통해 LLM을 최적화하고, 이를 기반으로 Intent Translation을 수행한다. 제안하는 방법을 기존 방법과 여러 LLM 모델을 비교하여 성능을 평가한 결과, 제안하는 방법이 정확도 측면에서 우수한 성능을 보였다.

Key Words : Intent Translation, Large Language Model (LLM), Intent-based Networking (IBN), NFV Management

ABSTRACT

As the scale and complexity of networks increase, the concept of Intent-based Networking (IBN) has been proposed to manage networks based on user intent. With the advent of Large Language Models (LLMs), Intent Translation, which apply LLMs to IBN and transform natural language intents into structured, abstracted forms that can be interpreted by the system, has been attracted. This paper proposes a method for generating templates that abstract and structure natural language intents to manage Network Function Virtualization (NFV) environment through Intent Translation. The proposed method optimizes an LLM using domain adaptation techniques such as Chain-of-Thought (CoT) and few-shot learning. We evaluate the proposed method by comparing it and existing methods. The experimental results showed that the proposed method can significantly enhance the translation accuracy.

※ 이 논문은 2025년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (RS-2024-00392332, 6G 네트워크 통합 지능평면 기술 개발).

• First Author : Pohang University of Science and Technology (POSTECH), Department of Computer Science Engineering, hosewq@postech.ac.kr

° Corresponding Author : POSTECH, Department of Computer Science Engineering, jwkhong@postech.ac.kr

I. 서론

오늘날 네트워크 가상화 (Network Virtualization) 기술의 발전과 cloud-native 인프라의 확산으로, 기존 전용 장비를 통해 폐쇄적으로 운용되던 네트워크 기능 (Network Functions)을 상용 서버에서 소프트웨어 형태로 운용이 가능하게 하는 NFV (Network Function Virtualization)는 오늘날 네트워크 관리 및 운용의 핵심 기술로 활용되고 있다.

이러한 NFV 환경은 네트워크 관리의 확장성과 유연함을 제공하였지만 네트워크 서비스가 다양해지고 수요가 증가함에 따라 네트워크 환경에서 다수의 VNF (Virtual Network Function)와 다양한 서비스를 운용하기 때문에, 네트워크 및 서비스의 배포, 스케일링, 보안/네트워크 정책 설정 등과 같은 관리 작업의 복잡도가 크게 증가한다 [1]. 그러나 여전히 네트워크 관리자 및 서비스 운영자는 이러한 관리 문제를 해결하기 위해 네트워크 관리 정책 또는 구성 (configuration)을 직접 관리하며, 이는 높은 전문성과 많은 수작업을 요구한다.

이러한 문제를 해결하기 위해, 인공지능 (Artificial Intelligence, AI)을 활용한 네트워크 관리 및 최적화 방법에 대한 연구와 intent를 기반으로 네트워크를 자동으로 구성하고 관리하는 Intent-based Networking (IBN) 개념이 제안되었다 [2]. 특히, IBN은 사용자가 요구하는 고수준 의도 (intent)를 기술하고, 시스템이 네트워크 관리 정책과 자원 구성을 자동으로 제공하는 것을 목표로 한다.

그러나 기존 접근 방법은 도메인별로 설계된 제한적인 Intent 언어 또는 고정 템플릿에 의존하거나, 규칙 (rule) 기반으로 파싱하여 저수준 정책 구성으로 변환하는 방식을 취하기 때문에, 다양한 시나리오나 자연어 기반의 사용자 intent를 유연하게 처리하기 어렵다.

한편, 대규모 언어 모델 (Large Language Model, LLM)의 발전으로 AI 분야에서 자연어 처리 능력이 향상되면서, 최근 LLM을 기반으로 프로그래밍 코드를 생성하거나 CLI와 같은 명령어, 또는 설정 파일을 자동으로 생성하는 연구들에 대한 시도가 증가하고 있다 [3]. LLM은 네트워크 관리 분야의 관련 용어와 개념을 이

해하고 예시 구성을 어느 정도 생성할 수 있지만, 자연어 intent로부터 네트워크 관리에 대한 정책이나 구성 생성 이전 단계인 Intent Translation 단계에서도 LLM의 hallucination으로 인해 잘못된 정책이나 구성을 생성한다.

본 논문은 자연어 intent에서 실행 가능한 정책 생성의 이전 단계인, 사용자의 자연어 intent를 추상화, 정형화된 중간 표현으로 변환하는 Intent Translation 단계를 다룬다. Intent Translation은 사용자가 자연어로 기술한 intent를 분석하여, NFV 환경의 관리 작업 유형 (Deployment, Service Function Chaining (SFC), Auto-scaling, Security Configuration, Network Configuration), 대상 인스턴스와 서비스, 자원 요구사항, 네트워크·보안 제약, Service Level Objective (SLO) 정보를 포함하는 JSON 기반 템플릿 (translated intent)을 생성한다. 이 translated intent는 이후 정책 생성 단계에서 실제 실행 가능한 네트워크 관리 정책을 생성할 때, 관리 및 운용하는 서비스가 정의된 Network Service Descriptor (NSD) [4]와 유사하게 intent의 대상 정책 및 자원에 대한 정의를 참조하는 템플릿으로 사용될 수 있으며, 자연어 이해와 정책 생성 과정을 분리함으로써 다양한 NFV 환경에 대한 확장성, 견고성과 재사용성을 높일 수 있다.

논문의 주요 기여는 다음과 같다:

- 네트워크 관리 작업을 표현하기 위해 관리 유형, 대상, 자원 요구, 제약 조건 등을 포함하는 JSON 형식 기반 intent 추상화 및 정형화 템플릿을 정의하는 Intent Translation 방법을 제안한다.
- 자연어 Intent가 정책 생성 및 관리 단계에서 사용될 수 있도록 intent를 추상화 및 정형화하여 LLM 기반 Intent Translation 방법을 설계 및 구현하고, LLM의 도메인 최적화 기법을 사용하여 정확도를 향상시키는 방법을 제시한다.
- 다양한 유형의 네트워크 관리 작업으로 구성된 intent 데이터셋을 구축하고, 제안 방법과 기존 방법의 정확도 및 처리 시간 성능을 비교, 분석하여 제안하는 Intent Translation 방법을 정량적으로 검증한다.

II. 관련 연구

2.1. LLM 및 도메인 최적화 기법

LLM은 대규모 텍스트 코퍼스 (corpus)를 기반으로 사전 학습 (pre-training)되어 자연어 이해와 생성, 요약, 질의응답 등 다양한 자연어 처리 작업을 단일 모델로 수행할 수 있는 범용 모델로 발전해왔다. 최근 LLM은 단순 문장 단위의 응답 생성뿐 아니라, 문서 단위의 텍스트를 이해하고 분석하거나 복잡한 지시를 해석하고, 단계적인 추론을 수행하며, 멀티 모달리티 데이터를 지원하는 기능으로 확장되고 있다. 이에 따라 LLM이 네트워크 관리와 같은 분야에도 활용될 가능성이 높아지며 주목을 받고 있다. 특히, 사용자가 자연어로 표현한 요구를 분석하여 기계가 처리 가능한 형식으로 변환하거나, 로그 분석을 통해 네트워크를 모니터링하고 관리하는 작업은 LLM의 자연어 이해 및 생성 능력이 네트워크 관리 분야에 직접적으로 기여할 수 있는 대표적인 응용 사례로 볼 수 있다 [5].

그러나 LLM은 사전학습된 언어 패턴과 상식과 같은 일반적인 지식에는 강점을 가지는 반면, 네트워크와 같은 특정 도메인에 특화된 용어, 제약 조건, 출력 형식에 대해서는 사전학습된 지식이 부족하거나 기술에 대한 최신 정보 반영이 어렵기 때문에 ChatGPT와 같이 기본적으로 제공되는 LLM을 그대로 네트워크 사용하기 어렵다는 한계가 있다.

이를 보완하기 위해, LLM을 사용 목적에 맞게 도메인 적응 (domain adaptation) 또는 도메인 최적화하는 다양한 기법들이 제안되었다. 대표적으로, 추가적인 학습 없이 프롬프트의 설계를 최적화하는 프롬프트 엔지니어링 (Prompt Engineering) [6]에는 단계적 추론을 위해 프롬프트를 설계하는 Chain-of-Thought (CoT) [7], 입력 프롬프트 내에 관련 예시를 함께 제공하는 few-shot learning [8], 목표로 하는 도메인 데이터로 추가 학습을 통해 LLM의 파라미터를 전체적으로 재조정하는 미세 조정 (fine-tuning) [9], 그리고 추가적인 도메인 지식을 검색하는 Retrieval Augmented Generation (RAG) [10]의 활용, self-refinement, Reinforcement Learning from Human Feedback (RLHF)과 같이 사람의 개입이나 모델이 자체적으로 LLM의 출력을 조정하는 Feedback [11] 매커니즘을 예로 들 수 있다.

프롬프트 엔지니어링 기법은 모델 파라미터를 변경하지 않고도 프롬프트 설계를 통해 도메인 지식을 주입하거나 LLM을 목적에 맞추어

최적화를 할 수 있다는 점에서 별도의 학습 인프라나 자원을 요구하지 않는다. 특히, 도메인에 특화된 작업을 수행하기 위해 프롬프트를 설계하거나 관련 예시를 프롬프트에 포함하면, LLM이 해당 도메인에 맞게 프롬프트를 기반으로 해석 및 추론하고, 그에 따른 출력을 생성하도록 유도할 수 있다. 반면, fine-tuning 기법은 도메인 데이터가 일정 규모 이상 확보되고, 추가 학습 환경을 구성할 수 있을 때 보다 안정적이고 일관된 성능을 기대할 수 있지만, 학습을 위한 초기 비용 등 오버헤드가 크다.

본 논문에서 다루는 Intent Translation은 자연어로 표현된 네트워크 관리 Intent를 JSON 기반의 translated intent라는 중간 표현으로 변환하는 작업이다. 따라서 본 연구에서는 대규모 파라미터 학습이 필요한 미세 조정보다는, CoT 기반 프롬프트 설계와 few-shot learning을 통한 도메인 최적화 기법에 초점을 맞추어 LLM을 Intent Translation 작업에 활용한다.

2.2. 관련 연구

IBN은 사용자의 고수준 요구사항을 기반으로 네트워크 관리 정책을 자동으로 생성하고 적용한다. 네트워크 규모와 복잡도가 증가함에 따라 네트워크 관리자의 부담이 커지고 있기 때문에, 이러한 관리 부담을 줄이기 위한 방법으로 IBN이 큰 관심을 받아왔다. 초기 IBN 연구들은 주로 규칙 기반 접근법을 사용하여 사전에 정의된 템플릿을 기준으로 intent를 해석하고, 네트워크를 구성하였다 [12]. 그러나 이러한 방식은 네트워크 환경의 동적 변화에 유연하게 대응하기 어렵고, 새로운 서비스나 정책을 반영할 때 확장성이 떨어진다는 한계를 가진다.

이를 보완하기 위해, 자연어로 표현된 intent를 정책이나 설정으로 변환하기 위해 Natural Language Processing (NLP)와 LLM을 활용하는 연구들이 제안되었다. 특히, LLM의 등장으로 자연어 기반 intent를 보다 정교하게 해석하고, 이를 정책이나 네트워크 설정 (configuration)의 생성 정확도를 크게 향상시킬 수 있을 것으로 기대된다.

이러한 흐름에서 LLM을 활용한 intent 기반 네트워크 관리 연구가 제안되고 있으며, 특히, Intent Translation에 초점을 두는 연구들이 다수 보고되고 있다. Dzevaroska et al. [13]은

OpenStack 환경에서 사용자의 자연어 intent를 인스턴스 배포 정책으로 변환하는 LLM 기반 시스템을 제안하였다. 이 연구에서는 LLM을 이용해 intent 분류 단계에서 사용자의 요구사항에 내포된 작업(task)과 동작(action)을 식별한 뒤, 이를 OpenStack API에 매핑되는 정책으로 변환한다. 하지만 해당 연구는 인스턴스 배포(deployment)와 상태 점검(health check) 기능에 한정되어 있으며, 오토스케일링, 보안 정책, 네트워크 설정과 같은 관리 기능은 다루지 못한다는 한계를 가진다.

Mekrache et al. [14]은 Code LLaMA와 few-shot learning을 사용하여 서비스를 배포하는 LLM 기반 방법을 제안하였다. 이 연구에서는 자연어 intent를 NSD 템플릿으로 변환하는 Intent Translation을 수행한다. Wang et al. [15]는 B5G/6G 네트워크를 대상으로 LLM을 이용해 자연어 intent를 기계가 읽을 수 있는 정책으로 변환하는 개념적 설계를 제안하였다. 이 연구는 프롬프트 엔지니어링, RAG, fine-tuning을 활용하여 LLM이 실시간으로 네트워크 트래픽과 로그를 분석하고, 네트워크를 표현하는 AI 모델과 결합하여 의사결정을 위한 정책을 생성하는 방안을 제시한다. 그러나 이 연구는 개념 설계 수준에 머물러 구현 및 실험을 통한 검증은 제시하지 않는다.

Tu et al. [16]은 in-context learning, 특히 few-shot learning을 사용하는 LLM 기반 intent translation 방법을 제안하였다. 이 연구에서는 자연어 intent를 NFV 설정 템플릿으로 변환하며, OpenStack 환경에서 사용되는 NFV 설정 템플릿 (인스턴스 deployment, SFC)을 생성한다. Nam et al. [17]은 LLM을 이용한 VNF 생성 및 설치 방법을 제안하였다. 이 연구에서는 Method of Procedure (MOP) 문서를 활용하여 설치 가이드를 분석하고, OpenStack 환경에서 실행 가능한 VNF 설치 명령어 및 설정을 Python 코드 형태로 자동 생성한다.

위와 같은 관련 연구들은 공통적으로 자연어 기반 intent를 형식화된 템플릿이나 설정으로 변환하는 Intent Translation을 다루지만, 그 범위나 환경이 제한적이라는 한계가 있다. 무엇보다 대부분 OpenStack 또는 특정 5G 테스트베드 환경을 대상으로 인스턴스 배포, NSD 템플릿 생성, VNF 설치 명령 생성 등 일부 기능에 초점을 맞추고 있다. 반면, 본 연구는 오토스케일링, 보

안 설정, 네트워크 설정과 같이 NFV 관리에서 빈번하게 요구되는 다양한 관리 작업에 대한 intent translation을 수행한다. 또한, translated intent의 구조를 NFV 관리 전반을 포괄하는 공통 템플릿으로 체계적으로 정의하고, 다양한 관리 작업과 LLM 모델을 대상으로 정확도와 처리 시간을 비교 및 분석한 연구도 상대적으로 부족하다. 본 연구는 NFV 관리를 위한 정책 구성 관련 작업 (서비스 및 인스턴스 deployment, SFC 구성, auto-scaling, security/network configuration)을 포함하는 intent 데이터를 통해 JSON 기반 추상화 및 구조화 템플릿을 생성하는 방법을 설계하고, LLM 기반 Intent Translation 방법을 체계적으로 평가한다.

III. LLM 기반 Intent Translation 설계 및 구현

3.1. 개요

본 장에서는 NFV 환경에서 사용자의 자연어 기반 intent를 실제 실행 가능한 정책으로 변환하기 이전 구조화된 템플릿 형식으로 변환하는 LLM 기반 Intent Translation 방법을 설명한다. Intent Translation은 사용자가 입력한 관리 요구(intent)를 분석하여 관리 작업 유형, 대상 인스턴스, 자원 요구사항, 네트워크 및 보안 정책, SLO (Service Level Objective) 제약 등을 포함하는 JSON 객체 형식 기반의 translated intent로 변환한다. 이를 통해 자연어 수준으로 표현된 추상적인 요구를 정책 생성 단계에서 직접 활용 가능한 중간 표현으로 정규화하고, 이후 정책 생성 단계에서 실제 실행가능한 관리 정책을 생성할 때 참조할 수 있는 정형화된 템플릿을 제공한다.

3.2. Intent Data 및 출력 구조 정의

제안하는 Intent Translation의 입력은 사용자가 자연어로 기술하는 NFV 관리 정책 생성 작업에 대한 요청이다. 본 연구에서는 ETSI NFV IFA 및 SOL 표준 문서, 관련 연구, 실제 NFV 기반 서비스 시나리오를 참고하여 NFV 환경에서 반복적으로 수행되는 관리 작업을 정의한다. 이를 바탕으로 VNF 인스턴스 및 서비스 배포 (Deployment), SFC, Auto-scaling, Security 및 Network Configuration 5가지를 관리 작업으로 정의한다.

각 관리 작업에서 실제 운용 환경에서 사용되는 자연어 기반 intent 표현을 수집 및 생성하여 본 연구에서 사용하는 intent 데이터셋을 구성한다. 이때 특정 NFVI에 종속적인 문법 (Kubernetes YAML manifest, OpenStack HoT 등)은 배제하고, “무엇을 하고 싶은지” 사용자의 의도에 중점을 둔 문장으로 intent 데이터를 구성한다. 예를 들어, “Firewall, IDS, DPI 순서로 SFC를 구성하라”, “CPU 사용률이 90%를 초과하면 web-scale 서버에 Auto-scaling 정책을 적용하라”, “제공 중인 특정 서비스 대해 IP 주소 10.10.10.16 및 포트 80을 차단하라”와 같은 NFV 환경을 전반적으로 관리하기 위한 문장이 이에 해당한다. 이러한 intent 데이터의 생성에는 서로 다른 표현, 동의어, 문장 구조를 일부러 섞어서 다양한 자연어 입력에 대해 일반화 성능을 확보한다.

Intent Translation의 출력인 translated intent는 JSON 객체로 표현되는 템플릿 구조를 갖는다. 이 템플릿은 크게 관리 작업 유형 (management task), 수행 동작 (action), 대상 인스턴스 및 리소스 (targets), 자원 요구사항 (requirements), 운영 및 보안 제약조건 (constraints)으로 구성된다.

Deployment 작업의 경우 대상 인스턴스 이름, 이미지, CPU/메모리 등 자원 요청 및 제한 등의 필드를 포함하고, SFC 작업에서는 체인을 구성하는 VNF 인스턴스의 순서와 각 인스턴스의 역할, 연결 방식, 대상 네임스페이스 등의 정보가 포함된다. Auto-scaling 작업에서는 대상 Deployment, 스케일링 조건, 최소/최대 인스턴스 수 등이 포함되며, Security/Network Configuration에서는 정책이 적용되는 대상 서비스 또는 네트워크와 허용 및 차단할 IP 주소, 포트 등 세부 파라미터가 포함된다.

Translated intent 템플릿은 관리 작업 종류에 상관없이 공통된 상위 필드 구조 (task, action, targets, requirements, constraints)를 유지하면서, 각 작업에 특화된 하위 필드를 계층적으로 포함하도록 설계한다. 누락된 정보는 기본값으로 채우고, 잘못된 단위나 표현은 정규화함으로써 자연어 기반 intent 문장의 모호성을 줄일 수 있다. 이와 같이 설계된 구조화된 템플릿을 기반으로 실제 정책 생성 단계에서 템플릿 구조를 참조하여 대상 NFVI에 대한 관리 정책을 생성할 수 있도록 한다.

기존 IBN 및 intent-driven networking 연구에서는 정형화된 텍스트 입력이나 제한된 슬롯 기반 양식을 사용하여 의도를 파싱하는 규칙 기반 또는 heuristic 접근법이 주로 사용된다. 그러나 이러한 방법은 자연어 표현 방법이 달라져도 인식을 실패하거나, 추상적인 관리 목적을 세밀한 자원 및 정책 요구사항으로 해석하는데 한계가 존재한다. 따라서 본 연구에서는 LLM의 자연어 이해 능력을 기반으로 단계별 추론 기반 프롬프트 설계 (CoT), 예시 기반 학습 (few-shot learning)을 적용한 프롬프트 엔지니어링을 사용하여, 다양한 자연어 표현으로 구성된 NFV 관리 intent를 정형화되고 추상화된 translated intent 템플릿으로 자동 변환하는 방법을 제안한다.

3.3. LLM 기반 Intent Translation 설계

제안하는 Intent Translation 방법은 IETF NMRG RFC 9316 [18]에서 정의한 intent 분류 방법론을 기반으로 설계한다. RFC 9316에서는 intent 기반 관리에서 사용자의 의도를 분석하는 절차를 intent solution, user types, intent type, intent scope, network scope, abstraction 등으로 구분하여 정의한다.

본 연구에서는 이 중 각 관리 요청에 대해 intent type, intent scope, network scope를 식별한 뒤, NFV 관리 도메인에 적합한 수준으로 의도를 추상화 (abstraction)까지의 범위를 대상으로 제안하는 Intent Translation 방법을 설계하고 구현한다.

LLM 기반 Intent Translation은 그림 1과 같이 설계된다. 먼저, 사용자의 자연어 intent를 입력

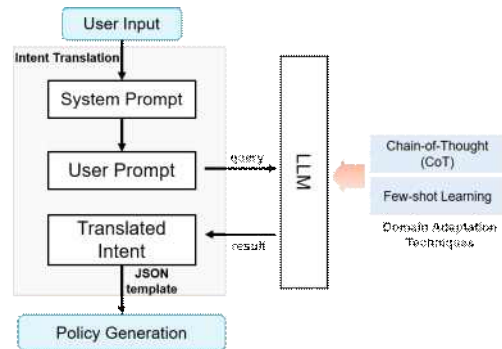


그림 1. NFV 관리를 위한 LLM 기반 Intent Translation 설계
Fig. 1. Design of LLM-based Intent Translation for NFV Management

받으면 CoT 기반 프롬프트 설계를 적용하여 구성된 System Prompt를 통해 LLM이 입력된 intent가 어떤 관리 작업 (tasks)에 해당하는지 (Deployment, SFC, Auto-scaling, Security Configuration, Network Configuration), 어떤 동작 (actions)을 수행해야 하는지 (create, update, delete), 어떤 리소스와 네트워크 범위를 대상으로 하는지 (대상 서비스 이름, 대상 인스턴스 이름 등)를 단계적으로 추론하도록 유도한다. 이때 System Prompt는 RFC 9316에서 정의된 Intent Classification 방법을 참고하여, LLM이 각 관리 intent의 유형과 범위를 명시적으로 분리해서 판단하도록 프롬프트 규칙을 서술한다.

다음으로, User Prompt에는 각 관리 작업별 설계된 Use-case 예시 (few-shot examples)를 포함한다. User Prompt는 각 작업마다 1개의 예시를 정의하고, 각 예시는 자연어로 표현된 intent와 해당 intent에 부합하는 translated intent (JSON 객체) 쌍으로 구성된다. 예를 들어 Deployment 작업은 웹 호스팅 서비스 배포, 특정 VNF 인스턴스 배포와 같은 시나리오를 포함한 JSON 템플릿으로 구성되고, SFC 구성은 SFC 구성 순서, SFC를 구성하는 인스턴스 요구 사항 등이 포함된다.

이와 같은 CoT와 few-shot 기반 프롬프트 엔지니어링 설계의 목적은, LLM이 자연어 intent를 단순히 요약하거나 재서술하는 것이 아니라, NFV 관리 도메인에서 의미 있는 필드들 (작업 유형, 대상, 자원/네트워크/보안 요구사항, 제약 조건 등)을 구조화된 템플릿 형태로 추출하도록 유도하는 데 있다. 또한 intent type과 scope, network scope를 명시적으로 식별하게 함으로써, 향후 실제 정책 생성 관련 연구 단계에서 intent가 목표로하는 네트워크 및 자원에 대한 정책을 생성해야 하는지를 일관성 있게 해석할 수 있도록 한다.

3.4. LLM 기반 Intent Translation 구현

상기 LLM 기반 Intent Translation 설계를 통해 구현된 Intent Translation을 위한 LLM 기반 파이프라인은 그림 2와 같다. 먼저, 프롬프트 엔지니어링 기법을 통해 정의된 System Prompt와 User Prompt를 로드하고, 사용자가 입력한 자연어 intent를 user prompt에 삽입하여 System Prompt와 User Prompt가 결합된 최종 프롬프트

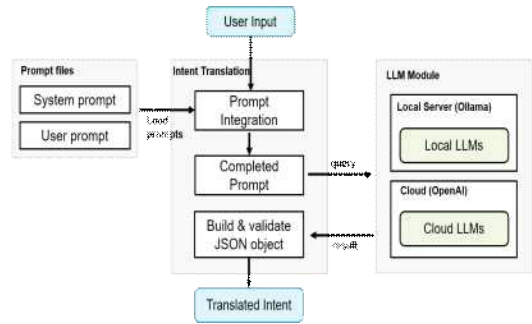


그림 2. LLM 기반 Intent Translation 파이프라인 구현
Fig. 2. Implementation of LLM-based Intent Translation

를 구성한다. System Prompt는 CoT를 기반으로 설계된 정의 및 규칙을 포함하여 LLM이 생성해야 할 JSON 객체의 구조와 단계별 추론 과정을 정의한다. User Prompt에는 관리 작업마다 1개의 few-shot 예시가 포함된다.

프롬프트가 구성되면, Intent Translation은 LLM 모듈에 프롬프트를 기반으로 요청을 전달하여 LLM이 생성한 응답을 받는다. 본 연구에서는 Ollama [19] 플랫폼을 사용하여 로컬 서버에서 실행되는 LLaMA 3.3 (70b) 모델을 사용하였으며, 로컬 환경에서 운용 가능한 LLM 모델과 클라우드 환경에서 운용 가능한 대형 LLM 모델의 성능을 4장에서 비교하기 위해 OpenAI [20] 플랫폼에서 제공하는 API를 사용하여 GPT-4o 모델을 사용한 파이프라인도 구현한다.

LLM 모델의 설정은 일관된 응답 생성을 위해 temperature, 최대 토큰 수, 컨텍스트 길이 등의 값을 설정하였다. 상기 로컬 및 클라우드 기반 LLM 호출 프로세스의 구현에는 LangChain [21]을 사용하여 프롬프트 구성부터 LLM 요청, 응답 수신을 위한 파이프라인을 구성하였고, 해당 파이프라인은 LLM이 생성한 결과물을 기반으로 JSON 객체를 생성하는 과정을 포함한다.

LLM이 생성한 응답은 때때로 설계 의도와는 다르게 자연어 설명 (instruction)과 JSON 객체 코드가 섞인 형태의 응답을 출력할 수 있다. 따라서 본 연구에서 제안하는 Intent Translation의 구현에는 설계 의도에 맞는 intent가 추상화된 JSON 템플릿 출력을 위해 LLM의 출력을 기반으로 Post-Processing를 통해 JSON 객체를 생성하고 검증하는 단계를 거쳐 최종 출력인 translated intent를 정제한다.

이 단계에서는 먼저 LLM의 출력을 기반으로

JSON 객체만을 추출하고, 코드 설명과 같이 JSON 객체 생성에 불필요한 텍스트를 필터링한다. 이후 필수 필드(task, action, targets, requirements, constraints 등)의 존재 여부와 계층적 필드 구조의 유효성을 점검하여 구조적 무결성을 검증한다. 검증에 실패할 경우에는 예외 처리 후 재생성을 요청하고, 디버깅 목적의 로그로 저장한다. 검증을 통과한 JSON 객체는 최종 출력으로 채택되어 JSON 템플릿 파일로 저장된다.

IV. 실험 및 검증

4.1. 실험 환경 및 구성

제안하는 Intent Translation 방법에 대한 검증 및 평가를 위해 실험 환경을 구축한다. 실험 환경은 3장에서 구현한 Intent 데이터셋과 Intent Translation 파이프라인을 사용한다.

Intent Translation은 자연어 intent를 JSON 템플릿으로 변환하는 단계이므로, 변환된 translated intent가 NFVI에서 어떤 자원과 정책으로 매핑되는지는 실제 정책 생성 단계에서 다뤄지기 때문에 OpenStack, Kubernetes와 같은

실제 NFV 환경 구축을 통한 검증 및 평가가 아닌 생성한 Intent 데이터셋을 기반으로 LLM의 생성 정확도를 평가한다. 이를 위해 실험에서는 translated intent가 실제 테스트베드 상의 관리 작업 (Deployment, SFC, Auto-scaling, Security, Network Configuration)으로 자연스럽게 대응되도록 intent 데이터와 템플릿 구조를 설계하여, 실제 네트워크 운용 시나리오와 괴리되지 않도록 데이터셋을 구성하고 Intent Translation 성능을 평가한다.

Intent Translation 성능 측정에는 두 종류의 자연어 기반 intent 데이터셋을 사용하였다. 첫 번째 데이터셋은 이전에 수행된 Intent Translation 선행연구 [16]에서 OpenStack 기반 NFV 환경을 대상으로 사용한 40개의 intent 데이터셋으로, Deployment와 SFC 두 가지 관리 작업만을 포함한다. 이 데이터셋은 제안하는 Intent Translation 방법과 이전 연구, 그리고 성능 평가를 위해 규칙 기반 heuristic 방법을 통해 구현된 Intent Translation 직접 비교하기 위해 사용된다. 두 번째 데이터셋은 본 연구에서 대상으로 하는 NFV 관리 작업을 기반으로 62개로 구성된 intent 데이터셋으로, 해당 데이터셋은 Deployment, SFC, Auto-scaling, Security, Network Configuration

표 1. Intent Translation 실험 평가에 사용된 LLM 모델
Table 1. Detailed LLM models for evaluation of Intent Translation experiment

LLM	Model Names (Parameters)	Used Model Descriptions
Mistral [22]	mistral:7b (7b)	- Mistral AI (2023.09) - Lightweight model
Phi4 [23]	phi4-reasoning:plus (14b)	- Microsoft (2025.04) - Lightweight, reasoning model
GPT-OSS [24]	gpt-oss:20b (20b)	- OpenAI (2025.08) - Open-weight model of GPT, support reasoning
Gemma3 [25]	gemma3:27b (27b)	- Google (2025.03) - Multi-modal/lingual model
Qwen3 [26]	qwen3-coder:30b (30b)	- Alibaba (2025.04) - Optimized for coding tasks, support reasoning
DeepSeek [27]	deepseek-r1:70b (70b)	- DeepSeek (2025.01) - High-capacity open model, support reasoning
LLaMA3.1 [28]	llama3.1:70b (70b)	- Meta (2024.07) - High-capacity open model, support reasoning
LLaMA3.3 [29]	llama3.3:70b (70b)	- Meta (2024.12) - Improved performance compared to llama3.1
GPT-4o [30]	gpt-4o (200b, estimated)	- OpenAI (2024.05) - Closed-weight model - Support full-cot and multi-modal reasoning

5가지 관리 작업마다 10개, combined task 12개로 구성된다. 이 데이터셋은 이전 연구에서 생성된 데이터셋보다 다양한 작업 유형과 난이도 수준에서 Intent Translation의 성능을 평가하기 위한 용도로 사용된다.

실험에 사용된 LLM 모델은 모두 Intent Translation에 동일한 형태의 프롬프트를 입력받도록 구성한다. 먼저, 첫 번째 실험에서는 제안 방법과 이전 연구의 Intent Translation 정확도 및 처리시간의 비교를 수행하며, 이를 위해 LLaMA 계열 모델을 사용한다. 이전 연구에서는 LLaMA 3.1 (70b) 모델을 사용하였고, 본 연구에서는 LLaMA 3.3 (70b)을 기본 LLM 모델로 사용하여 성능을 측정한다. 이후 실험에서는 Mistral, Phi4, GPT-OSS, Gemma3, Qwen3, LLaMA3.3, DeepSeek, GPT-4o 등 다양한 규모와 특성을 갖는 8종의 LLM 모델에 대해 동일한 Intent Translation 프롬프트를 적용하여, 모델 종류와 크기에 따른 성능 변화를 측정한다. 실험에 사용된 모델들의 세부 정보는 표 1과 같다.

성능 검증은 로컬 서버에서 실행이 가능한 LLM 모델은 Ollama 플랫폼을 통해 구축된 로컬 서버에서 수행하고, GPT-4o와 같이 공개되어 있지 않고 로컬 서버에서 실행이 불가능한 클라우드 기반 모델은 OpenAI API 호출을 통해 응답 생성을 요청하여 성능 검증을 수행한다. 각 모델은 최대 컨텍스트 길이를 동일하게 설정하고, 실험의 재현성과 일관된 비교를 위해 temperature는 0.0으로 고정한다.

4.2. 실험 1 - 기존 접근법과의 성능 비교

실험 1에서는 규칙 기반 heuristic 접근법을 사용하여 구현한 Intent Translation, 선행연구,

baseline LLM, 제안하는 방법으로 구현한 Intent Translation의 성능을 비교한다. 성능 비교를 위해 데이터셋은 이전 선행연구에서 사용한 데이터셋과 본 연구에서 생성한 데이터셋을 함께 사용한다. 먼저, 선행연구와 본 연구의 성능 비교를 위해, 이전 연구에서 사용한 40개의 intent 데이터셋 (Deployment 및 SFC 작업)을 그대로 사용한다. 이 데이터셋에 대해서는 규칙 기반 방법, 선행연구 (LLaMA3.1, LLaMA3.3), 제안하는 방법 (LLaMA3.1, LLaMA3.3)의 정확도와 처리 시간을 측정한다. 다음으로, 본 연구에서 생성한 62개의 intent 데이터셋 (5가지 관리 작업)을 대상으로 정확도 및 처리 시간을 측정하여 성능을 평가한다.

실험을 통한 정확도 및 처리시간 성능 측정 결과는 표 2와 나타난 바와 같다. 선행연구의 데이터셋을 통한 성능 측정 결과에서 규칙 기반 방법은 약 62.5% 수준의 정확도와 0.41초의 처리 시간을 보이며, 키워드를 중심으로 intent를 파악하고 해석하는 규칙 기반 접근법의 한계를 보였다. 도메인 최적화 기법을 적용하지 않은 기본 (baseline) LLaMA3.3 모델을 사용하여 Intent Translation을 수행했을 때에는 정확도가 75.5%로 향상되지만, 여전히 상당한 수의 펠드 누락 및 해석 오류가 발생하였다.

선행연구의 실험 결과는 9개의 few-shot 예시를 포함하는 few-shot learning 기법을 사용하여 LLaMA3.1에서 87.5%, LLaMA3.3에서 97.5%의 높은 정확도를 달성하지만, 처리 시간은 각각 6.91초, 7.20초로 증가하였다.

같은 데이터셋에 대해 본 연구에서 제안하는 방법 (CoT + few-shot learning)을 적용한 결과,

표 2. 실험1 - Intent Translation 기존 접근법과의 성능 비교 (accuracy, processing time)
Table 2. Experiment 1 - Performance comparison for existing approaches and proposed method

Methods		Rule-based (Heuristic)	LLM-based				
			Baseline LLM (LLaMA3.3)	Tu et al. [16]		Proposed	
				LLaMA3.1	LLaMA3.3	LLaMA3.1	LLaMA3.3
Tu et al. [16] dataset (Dep. + SFC)	Acc.	62.5	75.5	87.5	97.5	90.0	97.5
	Proc. time	0.41	1.83	6.91	7.20	4.57	4.93
Our dataset (5 tasks)	Acc.	59.7	72.6	-	-	90.3	96.7
	Proc. time	0.48	2.16	-	-	5.09	5.51

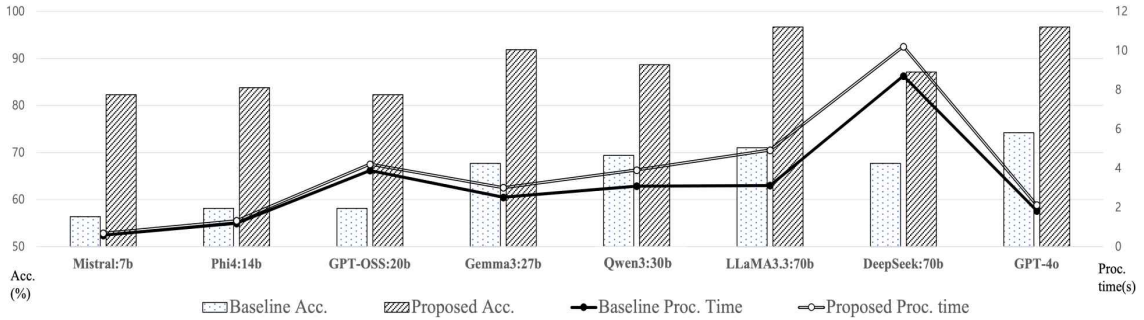


그림 3. LLM 모델별 Intent Translation 성능 측정 결과 (정확도, 처리 시간)

Fig. 3. Performance Comparison of Intent Translation by LLM models (accuracy, processing time)

LLaMA3.1에서는 약 90.0%, LLaMA3.3에서는 97.5%의 정확도를 달성하였다. 즉, 제안하는 방법은 선행연구와 비슷하거나 약간 높은 수준의 정확도를 유지하면서도 처리 시간은 각각 약 4.57초, 4.93초로 감소하였다. 이는 선행연구가 많은 수의 예시를 통해 추론 성능을 향상시킨 반면, 그에 따른 프롬프트 길이 증가로 인해 처리 시간이 크게 증가했다는 것을 의미한다. 반면, 제안하는 방법은 CoT 기반 단계별 추론과 필요한 수준의 예시만을 사용함으로써, 유사한 정확도를 유지하면서도 처리 시간을 줄일 수 있었다.

다음으로, 본 연구에서 생성한 62개의 intent 데이터셋에 대한 결과에서도 각 실험 결과는 유사한 경향을 보였다. 규칙 기반 Intent Translation 방법은 약 59.7%의 정확도와 0.48초의 처리 시간을 보였고, 기본 LLaMA3.3 모델은 약 72.6%의 정확도와 2.16초의 처리 시간을 보였다. 선행연구의 경우에는 Deployment와 SFC만을 대상으로 구현이 되어 있어 본 연구에서 생성한 데이터셋으로는 직접 비교가 불가능하기 때문에 해당 데이터셋에 대해서는 선행연구의 성능 측정이 진행되지 않았다.

제안하는 방법은 LLaMA3.1 기준 약 90.3%, LLaMA3.3 기준 약 96.7%의 정확도가 측정되었다. 이는 다양한 작업과 난이도가 섞여 있는 데이터셋에서도 제안하는 방법은 안정적인 성능 향상을 보였음을 의미한다. 처리 시간 측면에서는 제안하는 방법은 각 모델에서 약 5.09초, 5.51초로 증가했지만, 선행연구 대비하여 여전히 더 적은 처리시간을 보였다.

요약하면, 실험 1에서는 intent를 JSON 템플릿으로

추상화하는 작업에서 제안 방법이 규칙 기반 및 baseline LLM을 사용한 방법보다 높은 정확도를 보였고, 이전 연구와 비교하면 비슷한 수준의 정확도를 유지하면서도 처리 시간을 의미 있게 단축하는 결과를 보여주었다. 이는 NFV 관리 환경에서 다양한 자연어 기반 intent를 실시간에 가깝게 처리해야 하는 상황에서 제안 방법이 보다 실용적인 성능을 제공할 수 있음을 시사한다.

4.3 실험 2 - LLM 종류 및 크기에 따른 성능 비교

실험 2에서는 본 연구에서 제안하는 Intent Translation 방법이 다양한 종류와 크기를 가지는 LLM 모델 전반에서 일관된 성능 향상을 제공하는지 검증한다. 이를 위해 표 1에 설명된 Mistral (7b), Phi4 (14b), GPT-OSS (20b), Gemma3 (27b), Qwen3 (30b), LLaMA3.3 (70b), DeepSeek (70b), GPT-4o (200b 추정) 8종의 LLM 모델을 대상으로, 본 연구에서 생성한 62개 intent 데이터셋에 대해 baseline 방법과 제안하는 방법의 Intent Translation 정확도와 처리 시간을 측정하였다.

먼저, baseline LLM을 통해 Intent Translation의 성능을 측정한 결과, 파라미터 수가 적은 소형/경량화 모델은 Mistral, Phi4, GPT-OSS에서 각각 약 56.4%, 58.1%, 58.1%의 정확도를 보였으며, 약 30b의 파라미터를 사용하는 중형 크기의 모델인 Gemma3, Qwen3는 약 67.7%, 69.4%로 약 10% 높은 정확도를 보였다. 70b의 파라미터를 사용하는 상대적으로 대형 모델인 LLaMA3.3과 DeepSeek는 각각 71.0%, 67.7%의 정확도를 기록하였고, 클라우드 환경에서 API를 통해 수행되는 GPT-4o는 약 74.2%의 가장 높은 정확도를 보였다. 그러나 baseline LLM을 사용한 방법

은 관리 작업의 세부 필드나 제약조건을 반영하지 못하는 사례가 존재하여 단순 baseline LLM 방법만으로는 NFV 관리에 요구되는 수준의 Intent Translation 정확도를 보이지 못했다.

동일한 모델에 대해 제안 방법 (CoT + few-shot)을 적용한 결과, 모든 모델에서 정확도가 크게 향상되었다. 소형 모델의 경우 Mistral은 56.4%에서 82.3%로, Phi4는 58.1%에서 83.8%로 약 20~26%의 정확도 향상을 보였다. 중형 모델인 Gemma3와 Qwen3 역시 각각 67.7%에서 91.9%, 69.4%에서 88.7%로 성능이 크게 개선되었으며, LLaMA3.3은 71.0%에서 96.7%로 약 25%p 이상 향상되었다. DeepSeek는 67.7%에서 87.1%로, GPT-4o는 74.2%에서 96.7%로 향상되어, 클라우드 기반 대형 모델에서도 제안 방법이 높은 정확도 개선 효과를 제공함을 확인할 수 있었다.

처리 시간 측면에서는 프롬프트 복잡도 증가로 인해 모든 모델에서 소폭의 지연이 발생하였다. 예를 들어 Mistral은 평균 0.58초에서 0.68초로, Phi4는 1.19초에서 1.32초로 증가하였다. LLaMA3.3과 DeepSeek는 각각 약 1초 내외의 추가 지연이 발생하였고, GPT-4o는 약 0.3초 정도의 증가를 보였다. 그러나 해당 지연은 대부분 1초 이내 범위에 머물렀으며, 정확도 향상 폭을 고려할 때 Intent Translation 용도에서는 수용 가능한 수준의 오버헤드로 판단된다. 특히 소형·중형 모델의 경우 1초 미만의 처리 시간과 80~90% 이상의 정확도를 동시에 만족하여, 엣지 장비나 자원이 제한된 환경에서도 실용적으로 사용할 수 있는 가능성을 보여준다.

4.4 결과 분석

실험 1의 결과를 보면, 제안하는 LLM 기반 Intent Translation은 규칙 기반 방법과 baseline LLM에 비해 translated intent 생성 정확도에서 뚜렷한 우위를 보였다. 규칙 기반 방법은 처리 시간은 매우 짧지만 다양한 자연어 표현과 복잡한 intent로 인해 약 60% 정확도를 보인 반면, 제안하는 방법은 CoT 기반 단계적 추론과 예시를 통해 기존 LLM 기반 연구와 유사하거나 더 높은 정확도를 보였다. 특히, Deployment와 SFC에 한정된 선행연구의 데이터셋뿐 아니라 Auto-scaling, Security, Network Configuration까지 포함하는 62개 intent 데이터셋에서도 90% 이상의 정확도를 안정적으로 달성하여, NFV 관

리 작업이 확장된 환경에서도 높은 성능을 확보했음을 확인하였다.

실험 2에서는 제안하는 방법을 여러 종류의 LLM에 적용했을 때의 성능 변화를 분석하였다. Baseline LLM을 사용할 경우 모델 크기에 따라 약 50~70%의 정확도에 그쳤지만, 제안하는 방법을 적용하면 대부분의 모델에서 정확도 향상이 일관되게 관찰되었다. LLaMA3.3과 GPT-4o와 같은 대형 모델은 translated intent 템플릿의 구조와 제약조건을 비교적 정확하게 재구성하여 90% 중후반대의 정확도를 기록했으며, Mistral, Phi4 등 경량 모델도 80% 이상으로 상승해 자원이 제한된 환경에서도 활용 가능성을 보여주었다. 처리 시간은 프롬프트가 복잡해짐에 따라 baseline 대비 증가했으나, Intent Translation부터 정책 생성, 실제 배포까지 포함한 전체 NFV 관리 파이프라인 관점에서 보면 수십에서 수백초가 소요되는 전체 처리시간을 고려할 때 수용 가능한 수준의 오버헤드로 해석할 수 있다.

요약하면 실험 1과 실험 2의 결과는 Intent Translation 단계의 성능을 결정하는 주요 요인은 모델의 크기 등 파라미터 수가 아니라, CoT 프롬프트 설계 기반 단계별 추론과 few-shot learning과 같은 도메인 최적화 기법의 적용을 통해 LLM을 목표에 맞게 최적화하는 것임을 의미한다. 제안하는 방법은 다양한 모델과 관리 작업에 대해 intent를 추상화 및 구조화하여 재구성하며, 기존 규칙 기반의 접근법이나 baseline LLM을 사용하는 방법보다 처리 시간의 증가는 존재하지만 정확도 측면에서 우수하다. 향후에는 intent 데이터셋을 더욱 다양화하여 다양한 인텐트 유형과 자연어 표현에 대한 평가를 확대하고, 본 연구의 Intent Translation 방법을 기반으로 LLM이 실제 실행 가능한 관리 정책을 자동으로 생성하고 생성된 정책을 적용하는 NFV 정책 생성 및 intent 기반 자동화 관리 연구로 확장할 필요가 있다.

V. 결론

네트워크 가상화 기술이 발전함에 따라 NFV 환경의 네트워크 관리 복잡성이 증가하고 있다. 본 논문에서는 사용자의 자연어 관리 의도 (intent)를 기반으로 intent에 포함된 NFV 관리 정책 관련 작업

을 추상화 및 구조화하여 JSON 형식의 템플릿으로 변환하는 LLM 기반 Intent Translation 방법을 제안한다. 제안하는 방법은 CoT, few-shot learning을 사용하여 LLM 기반 파이프라인을 설계 및 구현하고, 기존 연구 및 여러 LLM 모델과의 성능 비교 실험을 통해 제안하는 방법이 기존 방법보다 높은 정확도를 보이고, 특히 구조가 복잡한 관리 작업에서 두드러진 성능 향상을 보이는 것을 확인하였다.

향후 연구에서는 intent 데이터를 다양하게 구성하여 NFV 서비스 유형과 장애 탐지/복구 [31-32], 자원 최적화 [33]와 같은 운영 시나리오까지 확장하여 제안하는 방법을 더욱 평가할 예정이다. 또한, 본 Intent Translation 연구를 확장하여 intent를 기반으로 Kubernetes [34], OpenStack [35]와 같은 실제 NFVI 운용 환경에서 실제 실행 가능한 정책을 생성하고 적용하는 Intent 기반 자동화 관리 프레임워크 개발에 대한 연구를 수행할 예정이다.

References

- [1] R. Boutaba et al., "A comprehensive survey on machine learning for networking: evolution applications and research opportunities", in *Journal of Internet Services and Applications*, vol. 9, issue 1, 2018.
- [2] Saha et al. "Intent-based networks: An industrial perspective", *Proceedings of the 1st international workshop on future industrial communication networks*. 2018.
- [3] Leivadeas et al., "A survey on intent-based networking", *IEEE Communications Surveys & Tutorials* vol. 25(1), pp. 625-655, 2022.
- [4] ETSI, "ETSI GS NFV-SOL 007, Network Functions Virtualisation (NFV) Release 3; Protocols and Data Models; Network Service Descriptor File Structure Specification", [Online]. Available at https://www.etsi.org/deliver/etsi_gs/NFV-SOL/001_099/007/03.05.01_60/gs_nfv-sol007v030501p.
- [5] Zhao et al., "A survey of large language models", *arXiv preprint, arXiv:2303.18223*, pp. 1-124, 2023.
- [6] L. Giray, "Prompt engineering with ChatGPT: a guide for academic writers", *Annals of biomedical engineering* vol. 51(12), pp. 2629-2633, 2023.
- [7] Wei et al., "Chain-of-thought prompting elicits reasoning in large language models", *Advances in neural information processing systems* vol. 35 pp. 24824-24837, 2022.
- [8] Song et al., "Llm-planner: Few-shot grounded planning for embodied agents with large language models", *Proceedings of the 2023 IEEE/CVF international conference on computer vision (ICCV 2023)*, pp. 1-12, 2023.
- [9] Lin et al., "Data-efficient Fine-tuning for LLM-based Recommendation", *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*, pp. 1-10, 2024.
- [10] P. Lewis et al., "Retrieval-augmented generation for knowledge-intensive nlp tasks", *Advances in neural information processing systems* 33 (NeurIPS 2020), pp. 9459-9474, 2020.
- [11] Stamper et al., "Enhancing llm-based feedback: Insights from intelligent tutoring systems and the learning sciences", *International Conference on Artificial Intelligence in Education*. Cham: Springer Nature Switzerland, pp. 32-43, 2024.
- [12] J. Hong et al., "A Comprehensive Survey on LLM-Based Network Management and Operations", *International Journal of Network Management (IJNM)*, vol. 35, issue 6, e70029, pp. 1-54, 2025.
- [13] Dzevaroska et al., "LLM-based policy generation for intent-based management of applications", *2023 IEEE 19th International Conference on Network and Service Management (CNSM)*, pp. 1-7, 2023.
- [14] Mekrache and Ksentini, "LLM-enabled intent-driven service configuration for next generation networks", *2024 IEEE 10th International Conference on Network Softwarization (NetSoft)*, pp. 253-257, 2024.
- [15] Wang et al., "Network meets ChatGPT: Intent autonomous management, control and operation", *Journal of Communications and Information Networks* vol. 8(3), pp. 239-255, 2023.

[16] Tu et al., "Intent-Based Network Configuration Using Large Language Models", International Journal of Network Management (IJNM), vol. 35, issue 1, 2024.

[17] S. Nam et al., "LLM-based VNF Deployment Automation in OpenStack Environment", 2025 IEEE/IFIP Network Operations and Management Symposium (NOMS 2025), Honolulu, USA, pp. 1-7, May 2025.

[18] IETF, "RFC 9316; Intent Classification", 2022. [Online]. Available at <https://www.rfc-editor.org/info/rfc9316>.

[19] Ollama, "Ollama Document", [Online]. Available at <https://docs.ollama.com/>.

[20] OpenAI, "OpenAI Documents", [Online]. Available at <https://platform.openai.com/docs/overview/>.

[21] Langchain, "Langchain Document", [Online]. Available at <https://docs.langchain.com/>.

[22] Mistral AI, "Mistral Document", [Online]. Available at <https://ollama.com/library/mistral/>.

[23] Microsoft, "Phi4 Document", [Online]. Available at <https://ollama.com/library/phi4-reasoning/>.

[24] OpenAI, "GPT-OSS Document", [Online]. Available at <https://ollama.com/library/gpt-oss/>.

[25] Google, "Gemma3 Document", [Online]. Available at <https://ollama.com/library/gemma3/>.

[26] Alibaba, "Qwen3 Document", [Online]. Available at <https://ollama.com/library/qwen3-coder/>.

[27] DeepSeek, "DeepSeek Document", [Online]. Available at <https://docs.langchain.com/deepseek-r1>.

[28] Meta, "LLaMA3.1 Document", [Online]. Available at https://www.llama.com/docs/model-cards-and-prompt-formats/llama3_1/.

[29] Meta, "LLaMA3.3 Document", [Online]. Available at https://www.llama.com/docs/model-cards-and-prompt-formats/llama3_3/.

[30] OpenAI, "GPT-4o Document", [Online]. Available at <https://openai.com/research/gpt-4o/>.

[31] 홍지범, 최민지, 유재형, 홍원기, "딥러닝 기반 네트워크 공격 및 침입 탐지 방법 연구", KNOM Review, Vol.25, No. 2, pp.10-21, Dec. 2022.

[32] 정세연, 홍지범, 유재형, 홍원기, "강화학습을 이용한 엣지 컴퓨팅 환경에서의 Live Migration 정책 제어 연구", KNOM Review, Vol. 25, No. 1, pp. 28-36, Aug. 2022.

[33] 이도영, 유재형, 홍원기, "NFV 환경에서의 Deep Q-Networks 기반 오토 스케일링 기술 연구", KNOM Review Vol. 23, No. 2, pp. 1-10, Dec. 2020.

홍 지 범 (Jibum Hong)



2017 한양대학교 ERICA

컴퓨터공학과 학사

2017 ~ 2020 포항공과대학교

컴퓨터공학과 석사

2020 ~ 현재 포항공과대학교

컴퓨터공학과 박사과정

<관심분야> SDN/NFV, 네트워크 모니터링, 네트워크 관리, 인공지능 응용

홍 원 기 (James Won-Ki Hong)



1983 Univ. of Western Ontario, BSc in Computer Science

1985 Univ. of Western Ontario, MS in Computer Science

1985 ~ 1986 Univ. of Western Ontario, Lecturer

1986 ~ 1991 Univ. of Waterloo, PhD in Computer Science

1991 ~ 1992 Univ. of Waterloo, Post-Doc Fellow

1992 ~ 1995 Univ. of Western Ontario, 연구교수

1995 ~ 현재 포항공과대학교, 컴퓨터공학과 교수

2007 ~ 2011 포항공과대학교, 정보통신대학원장

2007 ~ 2010 포항공과대학교, 정보통신연구소 연구소장

2008 ~ 2010 포항공과대학교, 컴퓨터공학과 주임교수

2008 ~ 2012 포항공과대학교, 정보전자융합공학부장

2012 ~ 2014 KT 종합기술원, 원장

<관심분야> 네트워크 트래픽 모니터링, 네트워크 및 시스템 관리, 블록체인, SDN/NFV