JKSCI

한국컴퓨터정보학회논문지
Journal of The Korea Society of Computer and Information
Vol. 31 No. 3, pp. 117-125, March 2026
https://doi.org/10.9708/jksci.2026.31.03.117

# Asymmetric Soft Prompt-RAG Interactions in Liquid vs. Transformer Lightweight LLMs: An Empirical Study on SBA Policy QA

Jun Oh Cheong*, Hyunchul Ahn**

*PhD Candidate, Graduate School of Business IT, Kookmin University, Seoul, Korea
**Professor, Graduate School of Business IT, Kookmin University, Seoul, Korea

## [Abstract]

This study examines how retrieval-augmented generation (RAG) and soft prompt tuning jointly influence performance and efficiency in lightweight large language models. Using an SBA policy QA benchmark, we compare a Liquid-based model and a Transformer-based model across 18 experimental configurations covering precision settings, RAG modes, and soft prompting. Results reveal architecture-dependent effects: soft prompting improves both answer quality and latency stability in the Liquid model, whereas it introduces a quality?efficiency trade-off in the Transformer model. Category-level analysis identifies numeric and criteria-based questions as persistent bottlenecks, highlighting the need for normalization-oriented improvements in lightweight LLM deployment.

▸ **Key words**: Lightweight large language models, Retrieval-augmented generation, Soft prompt tuning, 4-bit quantization, Liquid architecture, Transformer architecture

## [요 약]

본 연구는 경량 대규모 언어모델 환경에서 검색 증강 생성(RAG)과 소프트 프롬프트 튜닝이 성능과 효율성에 미치는 결합 효과를 분석한다. SBA 정책 질의응답 벤치마크를 활용하여, 정밀도 설정, RAG 방식, 소프트 프롬프트 적용 여부를 포함한 총 18개의 실험 구성 하에서 Liquid 기반 모델과 Transformer 기반 모델을 비교하였다. 분석 결과, 소프트 프롬프트의 효과는 모델 아키텍처에 따라 상이하게 나타났다. Liquid 모델에서는 응답 품질과 지연시간 안정성이 동시에 개선된 반면, Transformer 모델에서는 품질과 효율성 간의 뚜렷한 상충관계가 확인되었다. 또한 범주별 분석을 통해 수치 및 기준 기반 질문 유형이 지속적인 성능 병목 요인으로 나타났으며, 이는 경량 LLM 실무 적용을 위한 정규화 중심 개선의 필요성을 시사한다.

▸ **주제어**: 경량 대규모 언어 모델, 검색증강생성, 소프트 프롬프트 튜닝, 4비트 양자화, 리퀴드 기반 아키텍처, 트랜스포머 기반 아키텍처

• First Author: Jun Oh Cheong, Corresponding Author: Hyunchul Ahn
*Jun Oh Cheong (marucnc@kookmin.ac.kr), Graduate School of Business IT, Kookmin University
**Hyunchul Ahn (hcahn@kookmin.ac.kr), Graduate School of Business IT, Kookmin University
• Received: 2026. 01. 05, Revised: 2026. 01. 31, Accepted: 2026. 02. 20.


http://www.ksci.re.kr pISSN:1598-849X | eISSN:2383-9945

## I. Introduction

자원 제약이 존재하는 중소기업 환경에서 LLM(Large Language Model)을 서비스에 적용할 때 운영자는 정확도뿐 아니라 정답성 및 의미 일치로 대표되는 품질과 지연 그리고 자원 효율과 관련된 다목적 트레이드오프(trade-off)를 동시에 관리해야 한다[1]. 특히 정책이나 규정 그리고 매뉴얼 기반의 문서 QA 환경에서는 경량 LLM이 원문 근거를 안정적으로 활용하지 못하여 응답의 신뢰성이 저하되기 쉽다[2][3]. 이를 보완하기 위해 검색 증강 생성이 널리 활용되고 있으나 검색이나 리랭킹 그리고 컨텍스트 주입 과정이 추가되면서 지연과 자원 사용이 증가할 수 있으며 경량 LLM 환경에서는 품질 향상을 위한 RAG(Retrieval Augmented Generation)가 오히려 운영 가능성을 저해하는 역설이 발생할 수 있다[2][3]. 따라서 핵심 질문은 RAG 적용 여부를 넘어서 어떤 RAG 전략이 어떠한 조건에서 품질과 비용을 균형 있게 개선할 수 있는가로 확장된다.

한편 최근 모델 본체의 재학습 없이 도메인 적응을 수행하는 소프트 프롬프트(Soft Prompt) 기법이 주목받고 있다[4][5][6]. 소프트 프롬프트는 full fine-tuning 대비 학습과 배포 측면에서 부담이 작지만 경량 LLM과 양자화 그리고 RAG가 결합된 운영 조건에서는 그 효과가 항상 동일한 방향으로 나타나지 않는다[6]. 품질이 개선되더라도 출력 길이 증가나 디코딩 불안정으로 인해 지연이 악화될 수 있으며 반대로 과생성을 억제하면서 품질을 유지할 가능성도 존재한다[4][6]. 또한 Liquid 기반 모델과 Transformer 기반 모델처럼 아키텍처가 상이한 경우 소프트 프롬프트의 작동 방식과 효과 역시 달라질 수 있다[6]. 그럼에도 불구하고 기존 연구는 서로 다른 데이터셋과 RAG 구성 그리고 평가 지표를 혼합하여 비교하는 경우가 많아 관측된 성능 차이를 유발한 요인을 구조적으로 분리하여 해석하기 어렵다는 한계가 있다[2]. 특히 기존 연구는 서로 다른 데이터/파이프라인 조건을 혼합해 비교하거나 품질 지표 중심으로 보고되어, 경량 환경에서 아키텍처·정밀도·RAG·소프트 프롬프트 효과를 분리하고 운영 관점 결론을 내리기 어렵다.

본 연구는 이러한 한계를 보완하기 위해 동일한 데이터셋인 미국 중소기업청(SBA)의 정책 관련 질의응답(QA) 데이터와 동일한 RAG 파이프라인 그리고 동일한 평가 지표와 디코딩 및 재현성 가드 하에서 경량 LLM의 핵심 설계 요소가 품질과 지연 그리고 자원 효율에 미치는 영향을 실증적으로 분석한다[2]. 비교 대상은 Liquid 기반 LFM2-1.2B와 Transformer 기반 Qwen2.5-1.5B이며, 정밀도(bf16(BitFloat16)/nf4(NormalFloat4)),RAG (OFF/NAIVE/ADV), 소프트 프롬프트(nf4에서만 on/off)를 조합한 실험 매트릭스로 상호작용 효과를 정량화한다[2][7]. 평가는 F1, ROUGE-L뿐 아니라 SBERT(Sentence-BERT) 기반 의미 유사도와 운영 지표(지연 중앙값(p50), 최대 VRAM 사용량)를 포함하고, hit_cap 및 output_tokens 등 출력 안정성 지표를 함께 기록하여 '품질 향상'과 '운영 불안정'을 구분한다[8][9]. 또한 질문 유형을 definition/process /numeric/criteria로 구분해, 특히 numeric/criteria 병목을 식별하고 정규화/표준화 중심의 후속 개선 방향을 제시한다.

본 연구의 연구 질문은 다음과 같다.

RQ1. 동일 RAG 파이프라인 하에서, 경량 LLM 아키텍처(Liquid vs Transformer) 차이가 품질과 지연 그리고 자원 효율에 미치는 영향은 무엇인가?

RQ2. 소프트 프롬프트는 RAG 전략(OFF/NAIVE/ADV)과 결합될 때 품질과 지연 그리고 출력 안정성에 어떠한 상호작용을 유도하며 그 효과는 Liquid 기반 모델과 Transformer 기반 모델에서 어떻게 비대칭적으로 나타나는가?

따라서 본 연구는 경량 LLM 설계를 단일 성능 지표가 아니라 품질-지연-자원 효율(VRAM)을 함께 고려해야 하는 다목적 운영 설계 문제로 규정한다. 이를 위해 데이터셋, RAG 파이프라인, 디코딩/평가 가드를 동일하게 통제한 실험 프레임워크를 구성하고, 아키텍처·정밀도·Soft Prompt·RAG 전략의 결합이 문서 기반 QA의 운영 특성에 미치는 영향을 실증적으로 분석한다.

본 논문의 기여는 다음과 같다.

- 동일 통제 조건에서 Liquid(LFM2-1.2B)와 Transformer(Qwen2.5-1.5B)를 비교하여 아키텍처별 트레이드오프 특성을 정량화한다.
- nf4에서 소프트 프롬프트 on/off와 RAG (OFF/NAIVE/ADV)를 결합해 소프트 프롬프트-RAG 상호작용의 비대칭성을 품질 및 운영 지표(latency p50, VRAM peak, hit_ cap, output_tokens)로 제시한다.

이후 논문의 구성은 다음과 같다. II장은 관련 개념과 연구를 정리하고, III장은 실험 설계와 파이프라인 고정 정책을 기술한다. IV장은 실험 결과 및 트레이드오프 해석을 제시하며, 마지막 V장은 결론과 향후 과제를 제시한다.

## II. Preliminaries

### 1. Lightweight LLMs, Model Families, and Quantization

본 연구의 대상은 약 1~4B 규모의 초경량 LLM으로 단일 GPU 또는 제한된 VRAM 환경에서의 서비스 적용 가능성을 평가한다. 경량 LLM은 비용과 지연 측면에서 이점을 가지지만 문서 기반 질의응답 환경에서는 정답성과 근거 정합성 그리고 출력 안정성이 저하될 수 있어 RAG나 PEFT(Parameter Efficient Fine Tuning)와 결합되는 경우가 많다[3][4].

본 연구의 핵심 특징은 초경량 조건에서 모델 계열을 비교 변인으로 포함한다는 점이다. 구체적으로 Liquid 기반 LFM2-1.2B와 Transformer 기반 Qwen2.5-1.5B를 동일한 데이터와 동일한 파이프라인 하에서 비교하여 성능과 효율이 파라미터 규모뿐 아니라 아키텍처 특성에도 영향을 받을 수 있음을 전제로 분석한다[10][11]. 일반적으로 Transformer 기반 LLM은 self-attention을 중심으로 토큰 간 전역 상호작용을 모델링하며, 경량화 시에는 양자화/추론 최적화에 따라 품질-지연-메모리 간 절충 관계가 크게 달라질 수 있다. 반면 Liquid 계열은 동적 시스템 관점의 설계를 통해 상태 업데이트와 시간적 적응을 강조하는 접근으로 소개되며, 경량 운영에서의 효율 특성은 구현 및 모델 변형에 따라 상이할 수 있다[10][11]. 본 연구에서는 LFM2-1.2B를 Liquid 계열의 대표 사례로 선택하였다.

또한 실서비스 환경에서는 양자화가 널리 사용되므로 본 연구는 bf16과 nf4 정밀도를 비교한다. bf16은 품질 보존과 안정성 측면에서 유리한 반면 nf4는 VRAM 사용량을 크게 절감할 수 있으나 품질 민감도가 증가할 수 있다. 이에 따라 정밀도 변화가 RAG와 소프트 프롬프트의 효과와 어떻게 상호작용하는지, 그리고 그 양상이 두 모델 계열에서 동일하게 나타나는지 여부를 함께 관찰한다[4][7].

### 2. Retrieval-Augmented Generation and Mode Definitions(OFF / NAIVE / ADV)

RAG는 문서 코퍼스에서 근거를 검색해 입력 컨텍스트로 주입함으로써 환각을 완화하고 근거 기반 응답을 유도한다. 일반적으로 인덱싱-검색-reranking-컨텍스트 조립-생성/후처리로 구성되며, 경량 LLM 환경에서는 추가 단계가 지연과 자원 사용을 증가시킬 수 있다[3]. 본 연구는 동일 인덱스와 디코딩 설정을 유지한 채 RAG 전략을 3수준으로 정의한다.

- OFF: 검색·주입을 수행하지 않는 기준선[3].
- NAIVE: 검색 결과를 단순 규칙으로 결합해 주입(reranking/선택 정책 최소화)[3][12][13].
- ADV: reranking(예: cross-encoder)과 문서 쏠림 제한, 컨텍스트 길이 제약 등 고급 정책으로 근거 선택 정밀도 강화[3][14].

이는 RAG를 단일 기법이 아니라 품질-지연 목표와 결합된 설계 선택으로 비교하기 위함이다.

### 3. Soft Prompt and Efficiency Considerations

소프트 프롬프트는 자연어 프롬프트 대신 학습 가능한 연속 벡터(가상 토큰)를 입력에 주입하는 파라미터 효율적 적응 기법이다[4][5]. 본 연구는 기본 모델 가중치를 고정하고 가상 토큰 임베딩만 학습하는 프롬프트 튜닝 방식을 사용한다. 이는 full fine-tuning 대비 학습 파라미터 수가 작아 학습·저장·배포 오버헤드가 낮다[4][5].

효율 관점에서 소프트 프롬프트 비용은 학습과 추론으로 구분된다. 학습 단계에서는 업데이트 대상이 제한되어 빠른 적응이 가능하다[4]. 반면 추론 단계에서는 가상 토큰이 입력 길이를 증가시켜 계산량이 늘 수 있으나, 실제로는 출력 분포(출력 길이, 반복, 과생성) 변화가 총 토큰 수와 디코딩 지연에 영향을 미쳐 지연이 증가하거나 감소할 수도 있다[4]. 본 연구는 이를 검증하기 위해 품질 지표와 함께 output_tokens, hit_cap, tokens/sec, latency p50을 기록해 소프트 프롬프트의 효율 효과를 구조적으로 분석한다[4][9].

### 4. Related Work, Limitations, and Distinction

기존 연구는 양자화와 온디바이스 LLM 운영 측면에서 4-bit 계열 정밀도에서의 효율을 보고하였으나, 서비스형 문서 QA에서 RAG 및 소프트 프롬프트와 결합될 때의 출력 안정성 및 지연 변화까지 통합적으로 다루는 경우는 제한적이다[1][7]. RAG 연구는 reranking 및 컨텍스트 제약을 통해 품질 개선을 보고하지만, 경량 모델 환경에서는 추가 단계가 지연·자원 비용을 유발하여 운영 가능성과의 트레이드 오프가 발생할 수 있다[2][3][14]. Soft Prompt는 적은 파라미터로 도메인 적응이 가능하나, 양자화·RAG가 결합된 조건에서 효과가 항상 단조적으로 나타나지 않으며 출력 길이/반복/상한 도달 등 출력 분포를 통해 지연이 악화 또는 개선될 수 있다[4][5][6]. 요컨대 선행 연구는 서로 다른 데이터셋·파이프라인·평가 설정을 혼합해 비교하는 경우가 많아, 아키텍처·정밀도·RAG·소프트 프롬프트의 주효과 및 상호작용을 구조적으로 분리하기 어렵다는 한계가 있다[2]. 본 연구는 동일 데이터셋과 동일 RAG 파

이프라인, 동일 디코딩/평가 가드를 고정한 통제 비교(controlled comparison) 방식으로 18개 조건을 교차 실험하여, Liquid vs. Transformer에서 소프트 프롬프트-RAG 상호작용의 비대칭성과 품질-지연-VRAM-출력 안정성 관점의 트레이드오프를 정량화한다[2][7].

## III. The Proposed Scheme

본 연구는 경량 LLM 환경에서 모델 아키텍처와 정밀도 그리고 RAG 모드와 소프트 프롬프트 사용 여부가 품질과 지연 그리고 VRAM 사용량과 출력 안정성에 미치는 영향을 동일 조건을 통제한 비교 설계 하에서 분해하여 분석하기 위한 실험 프레임워크를 제안한다. 핵심 원칙은 데이터셋과 코퍼스 그리고 인덱스와 프롬프트 그리고 디코딩과 평가 및 로깅 설정을 모두 고정한 상태에서 비교 변인만을 체계적으로 교체하여 관측된 차이를 변인 자체의 효과와 상호작용으로 해석 가능하게 하는 데 있다.

### 1. Reproducibility Policy and Environment Snapshot

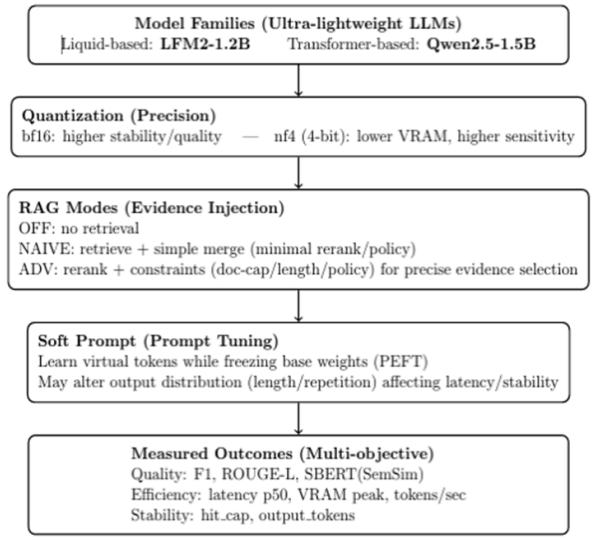


Fig. 1. Experimental design overview of quantization, RAG modes, and soft prompt tuning across Liquid and Transformer lightweight LLMs

Fig. 1에 제시된 바와 같이 재현성을 위해 다음 요소를 모든 실험 조건에서 동일하게 고정하였다: (1)데이터셋 버전 및 분할(SBA_2025-10-09_14-49-57), (2)인덱싱 및 검색 파라미터(청킹/Hybrid/윈도우/재랭커/컨텍스트 예산), (3) 프롬프트 및 디코딩 가드(단일 라인 출력, STOP 규칙, greedy 고정), (4) 평가 및 로깅 스키마(품질·시스템·안정성 지표 동시 기록), (5) 시드(42). 또한 실행 시점의 환경(하드웨어/소프트웨어 버전/경로)을 스냅샷으로 기록하여, 동일 조건 재실행 및 결과 추적을 가능하게 했다.

### 2. Experimental Setup and Factorial Design

태스크는 미국 SBA의 정책 문서를 근거로 하는 문서 기반 QA 문제로 설정하였으며 각 질의에 대해 근거에 기반한 단문 정답을 한 줄 형식인 Final: <answer> 형태로 출력하도록 구성하였다. 근거가 충분하지 않은 경우에는 Final: Unknown을 반환하도록 규정하여 환각 발생을 억제하였다.

비교 요인은 (1) 모델 아키텍처: Liquid 기반 LFM2-1.2B vs Transformer 기반 Qwen2.5-1.5B, (2) 정밀도: bf16/nf4, (3) 소프트 프롬프트: 0/1(단, nf4에서만 적용), (4) RAG 모드: OFF/NAIVE/ADV이다. 소프트 프롬프트가 nf4에서만 평가되므로 bf16에서 2×1×1×3=6 조건, nf4에서 2×1×2×3=12조건을 포함하여 총 18개 조건을 수행하였다(Fig. 2 참고).

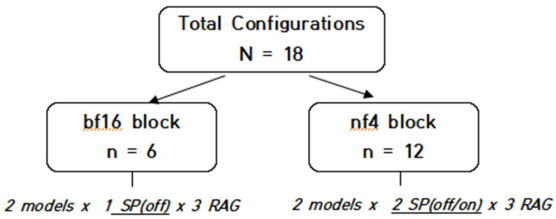


Fig. 2. Configuration count tree of the controlled experimental design

본 실험 설계는 파이프라인 변동을 최소화한 상태에서 주효과와 상호작용 효과를 분리하여 분석하기 위한 통제 비교 설계에 해당한다.

소프트 프롬프트는 nf4 조건에서만 on/off로 평가하였다. 이는 소프트 프롬프트 효과를 정밀도 변화(bf16↔nf4)와 분리하여, 동일 양자화 환경에서 소프트 프롬프트-RAG 상호작용과 아키텍처 비대칭성을 명확히 관찰하기 위한 설계이다. bf16+소프트 프롬프트 조합은 의미 있는 확장이나, 본 논문에서는 요인 증가에 따른 실험 규모/분량 제약으로 제외하고 후속 연구로 남긴다.

### 3. Data and Corpus Preparation

실험에 사용된 SBA 정책 QA 데이터셋은 전체 1,707개의 질의응답 쌍으로 구성된다. 이 중 1,391개는 학습 데이터로 316개는 검증 데이터로 사용하였다. 질문은

definition과 criteria 그리고 process와 numeric의 네 가지 카테고리로 라벨링되어 있으며 definition 642개와 criteria 448개 그리고 process 404개와 numeric 213개로 구성된다. 이러한 분류는 카테고리별 난이도와 오류 양상을 분리하여 분석하기 위한 기준으로 사용된다.

재현성을 확보하기 위해 데이터셋의 버전과 해시 정보 그리고 학습과 검증 분할과 시드 값을 모두 고정하여 결과 차이가 데이터 구성의 우연적 변동에서 비롯되지 않도록 통제하였다.

### 4. RAG Pipeline: Indexing, Retrieval, and Mode Definitions

문서 코퍼스는 정규화(공백/개행 처리) 후 고정 청킹 규칙으로 분절하였다(chunk_size=512, overlap=64, window_neighbors=3). 검색을 위해 sparse/dense 이중 인덱스를 구축하였다. BM25(Okapi)는 simple_tokenize_v2, k1=1.5, b=0.75를 사용하였고, dense 임베딩은 BAAI/bge-large-en-v1.5(1024d)로 생성한 벡터를 L2 정규화한 뒤 IP 기반 FAISS로 구성하였다.

Fig. 3에 제시된 바와 같이 Retrieval은 BM25와 dense 점수를 정규화한 뒤 가중 결합하는 hybrid 방식(α=0.40)으로 수행하며, 어휘 불일치를 완화하기 위해 RM3-lite 질의 확장을 적용한다. 또한 문맥 연속성 보완을 위해 인접 윈도우(window=3, weight=0.95, cap_delta=0.08)를 적용한다. 컨텍스트 조립은 과도한 장문 입력을 억제하기 위해 상한을 둔다(예: doc_cap=2, per_chunk_max=320, top_p_pages=8,ctx_char_limit=500, max_citations=2).

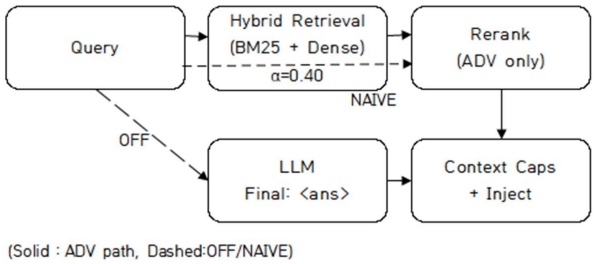


Fig. 3. Hybrid Retrieval and RAG Modes

RAG 모드는 동일 인덱스에서 주입 정책만 달리하도록 정의한다. OFF는 검색과 근거 주입 없는 모델 단독 응답이며, NAIVE는 하이브리드 상위 근거를 규칙 기반으로 결합한다. ADV는 후보 풀을 확장한 뒤 cross-encoder 재랭커(BAAI/bge-reranker-v2-m3)로 근거 선택 정밀도를 강화한다. 이때 일부 카테고리(numeric/criteria)에서 관찰되는 병목을 완화하기 위해 숫자, 단위, 섹션 힌트 기반의 제한적 부스팅/필터링을 적용하여 정답 스팬 포함 확률을 높였다.

### 5. Model Adaptation and Soft Prompt

소프트 프롬프트는 PEFT의 PROMPT_TUNING으로 구현하며, base 모델 가중치는 고정한 채 가상 토큰(virtual tokens)만 학습, 적용한다. 두 모델 모두 num_virtual_tokens=40, TEXT 초기화(prompt_tuning_init="TEXT")를 사용한다. Qwen2.5는 초기화 텍스트를 “Answer”로 설정하고 base_model_name_or_path를 명시하여 대상 모델 정합을 고정하였다. LFM2는 별도 체크포인트를 사용해 nf4 조건에서 소프트 프롬프트 결합 효과를 관찰한다.

### 6. Decoding Guards, Output Stability, and Evaluation/Logging

생성 단계에서는 평가 일관성과 서비스형 QA 요구사항을 위해 출력을 단일 라인(Final: <answer>)으로 제한하고, 근거가 부족한 경우 Final: Unknown을 반환하도록 규정하였다. 종료 마커는 suffix match 기반 stopping criteria로 적용한다. 디코딩은 greedy로 고정한다(do_sample=False, temperature=0.0, top_p=1.0, top_k=0). 또한 repetition_penalty=1.08, no_repeat_ngram_size=3, min_new_tokens=16을 동일하게 적용하였다. 최대 생성 토큰은 조건/카테고리별 상한으로 제한하며, 상한 도달 종료는 hit_cap으로 기록한다.

평가는 품질 지표(Exact Match(EM), token-level F1, ROUGE-L, SBERT cosine)와 시스템 지표(latency 및 p50, VRAM/ peak, tokens/sec, prompt/output tokens)를 동시 수집하고, stop_triggered, hit_cap, output_tokens 등 출력 안정성 지표를 함께 기록한다. 또한 numeric/criteria의 표기 변동 영향을 분리하기 위해 원문/패딩 기반 의미 유사도 값도 저장한다. 마지막으로 데이터셋 버전·해시·seed, 청킹 파라미터, α, 재랭커, 출력 형식과 디코딩 파라미터를 manifest로 고정하여 관측된 차이가 파이프라인 변동이 아닌 비교 변인 자체의 효과로 해석되도록 재현성 통제를 수행한다.

## IV. Experiments and Analysis

### 1. Performance, Cost and Output Stability

Table 1은 18개 조합에 대한 성능(품질/의미 유사도)

결과를 F1과 SBERT 기반 의미 유사도(SS)를 중심으로 제시하고 있다. 이는 엄격한 exact match 기준으로, 단답 QA에서의 표기 변동(특히 numeric/ criteria의 단위·기호·자리수·구두점 차이)에 민감하여 0/1로 급격히 변하는 경향이 있는 EM에 비해 두 지표가 모델 간 경향 비교에 보다 안정적인 장점이 있기 때문이다.

Table 1. Performance across 18 controlled configurations

| Model | Prec | SP | F1-O | F1-N | F1-A | SS-O | SS-N | SS-A |
|---|---|---|---|---|---|---|---|---|
| LFM2 | bf16 | 0 | 0.139 | 0.147 | 0.156 | 0.321 | 0.333 | 0.348 |
| LFM2 | nf4 | 0 | 0.171 | 0.159 | 0.174 | 0.435 | 0.390 | 0.394 |
| LFM2 | nf4 | 1 | 0.206 | 0.211 | 0.230 | 0.555 | 0.569 | 0.582 |
| Qwen2.5 | bf16 | 0 | 0.199 | 0.210 | 0.224 | 0.513 | 0.534 | 0.557 |
| Qwen2.5 | nf4 | 0 | 0.197 | 0.200 | 0.205 | 0.494 | 0.517 | 0.532 |
| Qwen2.5 | nf4 | 1 | 0.183 | 0.168 | 0.175 | 0.485 | 0.505 | 0.512 |

*Note. F1: token-level F1 score, SS: SBERT cosine similarity*

이 결과로 미루어 볼 때, 전체적으로 ADV는 근거 선택을 강화하여 품질을 개선하는 경향을 보였으며, 특히 LFM2에서는 nf4에서 소프트 프롬프트를 결합했을 때 OFF(O) → NAIVE(N) → ADV(A)의 단조 개선 패턴이 뚜렷하게 관찰되었다.

한편 Table 2는 ADV 모드에서 측정된 6가지 설정별 효율성 측정 결과를 제시하고 있다. 이 표를 통해 LFM2, nf4 그리고 소프트 프롬프트의 조합이 다른 그 어떤 조합보다 가장 우수한 효율을 보이고 있음을 확인할 수 있다.

Table 2. System cost and output stability (ADV mode only)

| Model | Prec | SP | p50(ms) | VRAMpk(MB) | OutTok | HitCap% |
|---|---|---|---|---|---|---|
| LFM2 | bf16 | 0 | 1,017.9 | 6,430 | 80.36 | 60.13 |
| LFM2 | nf4 | 0 | 1,106.7 | 6,516 | 78.16 | 72.15 |
| **LFM2** | **nf4** | **1** | **429.5** | **6,464** | **28.88** | **0.63** |
| Qwen2.5 | bf16 | 0 | 603.7 | 11,235 | 27.09 | 0.32 |
| Qwen2.5 | nf4 | 0 | 656.8 | 6,494 | 23.59 | 0.63 |
| Qwen2.5 | nf4 | 1 | 645.9 | 6,426 | 22.62 | 0.95 |

*Note. p50(ms): latency p50; VRAMpk(MB): peak VRAM; OutTok: mean output tokens; HitCap%: hit_cap rate. Rows 1-2 (LFM2, SP=0) have large p50 due to over-generation and cap hits (OutTok ↑, HitCap% ↑).*

ADV는 reranker와 컨텍스트 제약 등 복수 요소가 결합된 번들이며, 본 연구는 상호작용 분석을 위해 ADV 구성을 고정하여 비교하였다. 따라서 구성 요소별 기여도는 본 결과만으로 분리하기 어렵고, 다만 ADV에서의 품질 개선과 hit_cap/output length 감소 패턴은 reranker 기반 근거 선택 정밀도 향상의 기여 가능성을 시사한다. 다만 본 연구에서는 구성 요소별 기여도 정량화(reranker ON/OFF, 캡/쏠림 제한 단계적 추가)를 확인하지는 못했으며, 이는 후속 연구에서 제거 실험(ablation)으로 검증할 필요가 있다.

## 2. Soft Prompt and RAG Effects: Architecture-dependent Asymmetry

Fig. 4는 소프트 프롬프트의 효과를 시각적으로 제시하고 있다. 이 그림에서 y축을 구성하는 비율값(ratio)은 소프트 프롬프트 적용 이후, 각 성과지표가 증가한 정도를 나타낸다. 이 그림을 통해 알 수 있듯이, 소프트 프롬프트의 효과는 아키텍처에 따라 비대칭적으로 나타났다. LFM2(nf4, ADV)에서는 소프트 프롬프트 적용 시 품질(F1, Semantic Similarity(SemSim))이 완만하게 개선되었고 지연(p50)이 크게 감소하였다. 특히 동시에 hit_cap 및 output_tokens가 급감하고 있는데, 이는 "최대 토큰 상한에 의해 잘려 끝나는" 실패 모드가 제거되면서 출력 분포가 짧고 안정적으로 이동한 것이 지연 개선과 결합된 것으로 해석된다. hit_cap 감소는 절단 실패를 줄여 운영 안정성에 유리할 수 있으나, 과도한 출력 축소로 정보 누락/품질 저하가 발생할 가능성도 있다. 하지만 본 연구에서는 hit_cap/output length가 F1/SemSim과 함께 동반하여 개선되고 있음이 확인되었기 때문에, 이는 유의미한 결과로 해석이 가능하다.

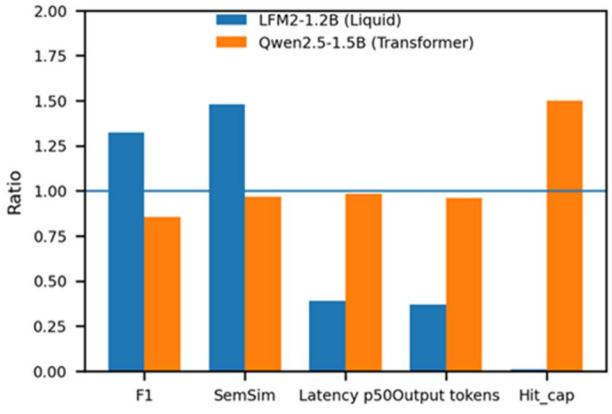


Fig. 4. Soft Prompt Effect under nf4 and ADV condition

반면 Qwen(nf4, ADV)에서는 소프트 프롬프트 적용 시 지연과 출력 길이는 소폭 감소했으나 품질이 하락하여, 효율 개선-품질 저하의 트레이드오프가 관측되었다. 또한 RAG 이득은 소프트 프롬프트와 상호작용하는 것으로 확인되었다. Table 1에서 알 수 있듯, LFM2에서는 소프트 프롬프트가 RAG(특히 ADV)의 이득을 강화하는 반면

Qwen에서는 RAG 이득이 부분적으로 상쇄되는 경향이 관찰되었다.

Qwen(nf4)+소프트 프롬프트에서의 품질 저하는 소프트 프롬프트가 항상 보편적 성능 향상으로 귀결되지 않음을 보여준다. 소프트 프롬프트는 입력 prefix로 작동하여 초기 hidden state와 attention 분포를 미세 조정하는데, nf4 양자화로 인한 표현 정밀도 저하가 이러한 조정을 불안정하게 만들 수 있다. 또한 Transformer 계열에서는 소프트 프롬프트 토큰이 RAG 컨텍스트와 입력/attention 예산을 경쟁하여 근거 토큰에 대한 집중을 약화시키는 방식으로 품질 저하가 나타날 가능성이 있다. 본 연구의 Qwen(nf4)에서는 output_tokens 및 지연이 소폭 감소하면서 품질이 하락하는 패턴이 동반되어, '과생성 억제-정답성 손상'의 트레이드오프 현상으로 해석될 수 있으며 이는 아키텍처·정밀도·RAG 결합 조건의 민감도를 시사한다.

### 3. Category-wise Bottlenecks and Output Stability

카테고리별 분석 결과, 병목 현상은 numeric/criteria에 집중되었다. LFM2에서는 소프트 프롬프트 적용 후 definition/process의 개선이 두드러졌고 numeric에서도 개선이 관찰되었으나 절대적인 성능 수준은 여전히 낮게 나타났다. 이는 단위와 기호 그리고 자리수와 구두점 등 표기 변동에 대한 정규화와 표준화가 후속 과제로 남아 있음을 시사한다. 반면 Qwen에서는 소프트 프롬프트 적용 시 definition과 process 유형의 성능이 오히려 취약화되는 경향이 확인되어, 비대칭적 효과가 설명형 및 절차형 질의에서 특히 두드러졌다.

운영 관점에서 가장 큰 변화는 LFM2(nf4)의 출력 안정성/토큰 효율 개선이다. 소프트 프롬프트 적용 시 hit_cap 비율이 크게 감소하고 출력 길이도 짧아져, 단순 tokens/sec의 개선보다 출력 분포 변화(과생성 억제)가 지연 개선을 견인했음을 짐작할 수 있다.

마지막으로 품질-지연-VRAM 관점에서 LFM2(nf4)+Soft Prompt(ADV)는 품질을 유지하면서 지연이 낮게 관측되어 파레토(Pareto) 관점의 유력 후보(frontier candidate)로 해석될 수 있다. 반면 Qwen(bf16, ADV)은 고품질이나 VRAM 비용이 증가하며, Qwen(nf4)+소프트 프롬프트는 효율이 소폭 개선되는 대신 품질이 저하되는 트레이드오프를 형성한다(Figs. 5-6 참고). 종합하면 소프트 프롬프트의 효과는 아키텍처 및 RAG 결합 조건에 의해 좌우되며, LFM2에서는 소프트 프롬프트가 RAG 이득과 출력 안정성을 동시에 강화하는 반면 Qwen에서는 효율 개선이 품질 저하로 교환되는 비대칭성이 나타났다.

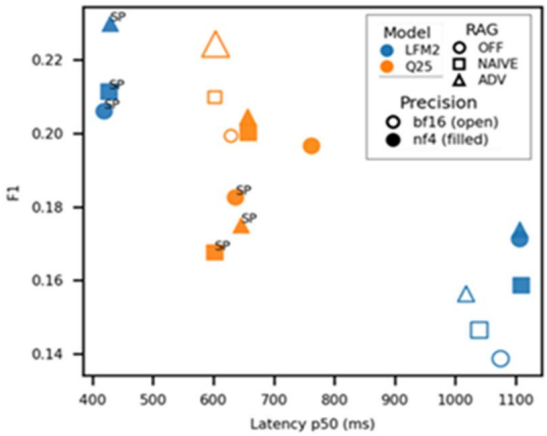


Fig. 5. F1-Latency p50 trade-off

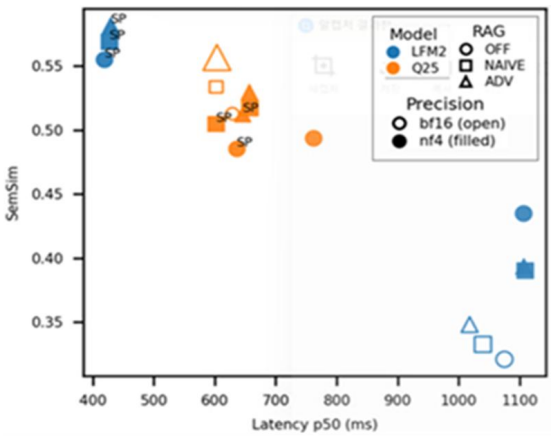


Fig. 6. SemSim-Latency p50 trade-off

## V. Conclusions

본 연구는 자원 제약이 존재하는 경량 LLM 환경에서 모델 아키텍처(Liquid vs. Transformer), 양자화 정밀도(bf16/nf4), RAG 전략(OFF/NAIVE/ADV), 소프트 프롬프트가 품질-지연-자원 효율(VRAM)에 미치는 영향을 실증적으로 분석하였다. SBA 정책 QA에서 LFM2-1.2B(Liquid)와 Qwen2.5-1.5B(Transformer)를 동일 데이터·동일 RAG 파이프라인·동일 평가/디코딩 가드 하에서 통제 비교하였다.

첫째, 동일 파이프라인 조건에서도 두 모델 계열은 품질·지연·VRAM 측면에서 상이한 트레이드오프 특성을 보였다. Qwen2.5의 bf16은 높은 품질을 제공하나 자원 비용이 증가할 수 있으며, nf4에서는 자원 이점에도 불구하고 품질·지연의 동반 개선이 항상 보장되지 않았다. 반면 LFM2는 nf4에서 소프트 프롬프트 및 RAG 결합 조건에 따라 품질과 운영 효율이 동시에 개선되는 구간이 관찰되

어, 경량 운영 시나리오에서 경쟁력 있는 설계 후보가 될 수 있음을 시사한다.

둘째, 소프트 프롬프트-RAG 상호작용은 아키텍처에 따라 비대칭적으로 나타났다. LFM2(nf4)에서는 소프트 프롬프트 적용 시 ADV에서 품질(F1/의미 유사도)이 개선되는 동시에 지연 p50이 크게 감소했고, hit_cap 및 output_tokens가 함께 낮아져 출력 안정성이 강화되었다. 또한 소프트 프롬프트 적용 이후 OFF → NAIVE → ADV로 갈수록 성능이 개선되는 경향이 확인되어, 소프트 프롬프트가 근거 활용 능력을 보완하며 RAG 효과를 활성화할 가능성을 보여준다. 반면 Qwen2.5(nf4)에서는 소프트 프롬프트 적용 시 지연 및 출력 길이가 소폭 감소하더라도 품질과 의미 유사도가 저하되는 트레이드오프가 관찰되었다.

종합하면, SBA 정책 QA에서 소프트 프롬프트-RAG 결합 효과는 경량 LLM 계열(Liquid vs. Transformer) 간 비대칭적 상호작용으로 관찰되었으며, LFM2에서는 RAG(특히 ADV) 하에서 근거 활용과 출력 안정성이 강화되는 반면 Qwen2.5에서는 동일 nf4 조건에서 품질-효율 trade- off가 나타났다. 학술적으로 본 연구는 동일 통제 설정에서 아키텍처×정밀도×RAG×소프트 프롬프트의 상호작용을 분리해 관찰하고(hit_cap, output length 등 출력 안정성 포함), '품질 향상'과 '운영 안정성/지연'의 연결을 분석 가능한 변수로 제시했다는 의의가 있다. 실무적으로는 자원 제약 환경에서 모델·정밀도·RAG·소프트 프롬프트 선택을 정확도 단일 기준이 아니라 품질-지연-VRAM-출력 안정성의 다목적 기준으로 결정해야 하며, 운영 단계에서 hit_cap과 output_tokens를 모니터링하면 과생성/상한 도달로 인한 지연 급증을 조기에 탐지·완화할 수 있다는 시사점을 제공한다.

본 연구의 한계는 다음과 같다. 첫째, 분석은 SBA 정책 규정 도메인에 기반하므로 절대 성능 수준과 오류 양상은 다른 문서 유형(기술/의료/법률)에서 달라질 수 있다. 둘째, 소프트 프롬프트는 실험 설계상 nf4 조건에서만 on/off로 비교되어 bf16+소프트 프롬프트 조합에 대한 결론은 본 연구 범위를 벗어난다. 셋째, ADV 모드는 reranker, 컨텍스트 캡, 문서 쏠림 제한 등 복합 요소로 구성되어, 본 결과만으로는 개별 구성요소의 기여도를 완전 분해하기 어렵다. 또한 본 연구는 SBA 정책/규정 도메인에 기반하므로 절대 성능 수준은 다른 문서 유형(기술/의료/법률)에서 달라질 수 있으나, 동일 파이프라인/가드 하에서 관찰된 비대칭 상호작용과 출력 안정성 변화(hit_cap, output length)가 지연에 미치는 영향은 문서 QA 전반에서 검증 가능한 운영적 메커니즘으로 해석된다. 향후 연구는 출력 안정성과 품질 간 연결 분석, 정규화/컨텍스트 구성의 비용-품질 민감도 평가와 함께, 서로 다른 도메인 코퍼스에서 동일 통제 설정으로 재현 실험을 수행하여 일반성과 도메인 의존 요인을 분리 검증할 필요가 있다. 아울러 정규화/표준화 외에도 answer post-processing 및 schema-based/constraint decoding을 결합해 형식 안정성과 numeric/criteria 정확도를 추가로 개선하는 방향에 대해서도 향후 추가적인 연구가 요구된다.

## REFERENCES

[1] X. Wang, Z. Tang, J. Guo, T. Meng, C. Wang, T. Wang, and W. Jia, "A Comprehensive Survey on On-Device AI Models," ACM Computing Surveys, Vol. 57, No. 9, Article 228, pp. 1-39, April 2025. DOI: 10.1145/3724420

[2] Y. Fan, X. Shen, Z. Zhu, R. Wang, W. He, and S. Wan, "A Survey on RAG Meeting LLMs: Towards Retrieval-Augmented Large Language Models," in Proc. ACM SIGIR, 2024. DOI: 10.1145/3637528.3671470

[3] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Kuttler, M. Lewis, W. Yih, T. Rocktaschel, S. Riedel, and D. Kiela, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," arXiv preprint, May 2020. DOI: 10.48550/arXiv.2005.11401

[4] B. Lester, R. Al-Rfou, and N. Constant, "The Power of Scale for Parameter-Efficient Prompt Tuning," in Proc. EMNLP, pp. 3045-3059, November 2021. DOI: 10.18653/v1/2021.emnlp-main.243

[5] X. L. Li and P. Liang, "Prefix-Tuning: Optimizing Continuous Prompts for Generation," in Proc. ACL, pp. 4582-4597, August 2021. DOI: 10.18653/v1/2021.acl-long.353

[6] N. Ding, Y. Qin, G. Yang, F. Wei, and B. Zhou, "Parameter-Efficient Fine-Tuning of Large-Scale Pre-Trained Language Models," Nature Machine Intelligence, Vol. 5, pp. 220-235, March 2023. DOI: 10.1038/s42256-023-00626-4

[7] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "QLoRA: Efficient Finetuning of Quantized LLMs," arXiv preprint, May 2023. DOI: 10.48550/arXiv.2305.14314

[8] C.-Y. Lin, "ROUGE: A Package for Automatic Evaluation of Summaries," in Proc. ACL Workshop: Text Summarization Branches Out, pp. 74-81, July 2004.

[9] N. Reimers, and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," in Proc. EMNLP-IJCNLP, pp. 3982-3992, November 2019. DOI: 10.18653/v1/D19-1410

[10] A. Amini, A. Banaszak, H. Benoit, et al., "LFM2 Technical

Report," arXiv preprint, November 2025. DOI: 10.48550/arXiv.2511.23404

[11] A. Yang, B. Yang, B. Zhang, et al., "Qwen2.5 Technical Report," arXiv preprint, December 2024. DOI: 10.48550/arXiv.2412.15115

[12] V. Karpukhin, B. Oguz, S. Min, et al., "Dense Passage Retrieval for Open-Domain Question Answering," arXiv preprint, April 2020. DOI: 10.48550/arXiv.2004.04906

[13] S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford, "Okapi at TREC-3," in Proc. The Third Text REtrieval Conference (TREC-3), NIST Special Publication 500-225, 1995.

[14] R. Nogueira, and K. Cho, "Passage Re-ranking with BERT," arXiv preprint, January 2019. DOI: 10.48550/arXiv.1901.04085

## Authors

Jun Oh Cheong received his Master's degree from the Graduate School of Business at Hankuk University of Foreign Studies, Korea, in 2022. He is currently pursuing a Ph.D. at the Graduate School of Business IT at Kookmin University. His primary research interests include LLM services of RAG and AI agent as well as AI-based business innovation strategies for SME.

Hyunchul Ahn received his B.S. degree in Industrial Management and his M.E. and Ph.D. degrees from the KAIST Graduate School of Management, South Korea. He is currently a Professor at the Graduate School of Business IT, Kookmin University. His research interests include AI applications in finance and marketing, as well as information systems adoption.