JKIICE Journal of the Korea Institute of Information and Communication Engineering

한국정보통신학회논문지 Vol. 30, No. 4: 591~596, Apr. 2026

# 의미 기반 동적 배칭과 캐싱을 통한 LLM 호출 최적화

김소룡[1] · 김명선[2*]

# Optimizing LLM Calls through Semantic Dynamic Batching and Caching

**Soryong Kim[1] · Myungsun Kim[2*]**

[1]Graduate Student, Department of Applied AI, Hansung University, Seoul 02876, Korea
[2]*Associate Professor, Department of Applied AI, Hansung University, Seoul 02876, Korea

## 요 약

대규모 언어 모델(LLM)을 활용한 실시간 대화형 서비스는 챗봇, AI 튜터, 지능형 에이전트 등 다양한 응용 분야로 확산되고 있으나, 높은 추론 비용과 응답 지연이 여전히 큰 제약 요인이다. 특히 표현은 다르지만 의미가 유사한 질의가 반복적으로 발생하고, 짧은 시간 구간에 유사 질의가 집중되는 사용자 패턴으로 인해 동일 혹은 유사 질의가 매번 독립적인 LLM 호출로 처리되면서 자원 낭비와 지연이 누적된다. 한편, 임베딩 기반 질의 유사도 검색을 활용하는 시맨틱 캐싱 기법은 과거 응답을 재사용하여 LLM 호출 빈도와 응답 시간을 줄이는 효과적인 방법으로 주목받고 있다. 본 논문은 이러한 시맨틱 캐싱을 보완하기 위해, LLM 호출 이전 단계에서 의미적으로 유사한 질의를 임베딩 공간에서 그룹화하는 의미 기반 동적 배칭 기법을 제안한다. bge-m3 임베딩과 코사인 유사도를 이용해 질의를 실시간으로 클러스터링하고, 대표 질의에 대해서만 캐시 조회 및 LLM 호출을 수행함으로써 LLM 기반 대화형 서비스의 호출 비용과 응답 시간을 동시에 절감할 수 있음을 실험을 통해 보인다.

## ABSTRACT

Real-time conversational services with LLMs are widely used in applications such as chatbots, AI tutors, and intelligent agents, but high inference cost and latency remain major bottlenecks. User queries in these services often show repetitive patterns where semantically similar intents are expressed in diverse forms and concentrated within short time windows, causing redundant LLM calls for identical or similar queries and accumulating resource waste and delay. Semantic caching based on embedding similarity search has emerged as an effective approach to reuse past responses and reduce the frequency and latency of LLM calls. To complement such caching, we propose a semantic dynamic batching method that groups semantically similar queries in the embedding space before invoking the LLM. By clustering queries in real time using bge-m3 embeddings and cosine similarity, and performing cache lookup and LLM inference only for representative queries, we experimentally demonstrate that the proposed method can simultaneously reduce invocation cost and response time in LLM-based conversational services.

**키워드** : 대규모 언어 모델, 의미 기반 동적 배칭, 시맨틱 캐싱, 임베딩, 코사인 유사도

Keywords : Large Language Model, Semantic Dynamic Batching, Semantic Caching, Embedding, Cosine Similarity

**Received** 24 January 2026, **Revised** 3 February 2026, **Accepted** 12 March 2026

*** Corresponding Author** Myung-sun Kim (E-mail : kmsjames@hansung.ac.kr Tel +82-02 -760-4045)
Department of Applied AI, Hansung University, Seoul 02876, Korea

Open Access http://doi.org/10.6109/jkiice.2026.30.4.591 pISSN:2234-4772

## Ⅰ. 서 론

대규모 언어 모델(Large Language Models, LLM)을 활용한 실시간 대화형 서비스는 챗봇, AI 튜터, 지능형 에이전트 등 다양한 응용 분야로 확장되고 있다[1,2]. 그러나 이러한 서비스는 LLM의 높은 추론 비용과 긴 추론 시간을 요구한다[3]. 특히 대규모 사용자 환경에서는 동일하거나 유사한 질의가 반복적으로 발생함에도 불구하고[4], 매 요청을 독립적인 LLM 추론으로 처리함에 따라 불필요한 계산 자원이 소모되는 문제가 두드러진다.

LLM 기반 대화형 서비스의 사용자 질의 패턴은 크게 두 가지 특징을 보인다. 첫째, 표현은 상이하나 의미적으로 동일한 질의가 반복되는 '중복성'이다[4,5,6]. 둘째, 짧은 시간 구간 내에 유사한 질의가 집중적으로 요청되는 '집중성'이다. 이러한 특징을 가진 다수의 유사 질의를 각각 독립적인 LLM 추론으로 처리할 경우, 반복적인 연산으로 인한 막대한 자원 낭비와 응답 지연을 초래한다[3].

이러한 비효율을 개선하기 위해 GPTCache[5]와 같은 시맨틱 캐싱(Semantic Caching) 연구가 제안되었으며, 이는 임베딩 유사도 검색을 통해 과거의 응답을 재사용함으로써 비용을 절감하였다[5,7]. 그러나 기존 방식은 시간차를 두고 발생하는 순차적 요청에는 효과적이지만, 특정 시점에 유사 질의가 몰리는 '연속 유사 요청' 상황에서는 요청마다 캐시 미스가 발생하여 각각 LLM을 호출하게 되는 한계가 있다[5,6].

본 연구에서는 시맨틱 캐싱 기법을 확장하여, 연속 유사 요청 상황에 대응하는 의미 기반 동적 배칭 기법을 제안한다. 제안 기법은 의미적으로 유사한 질의를 실시간으로 병합하여 단일 LLM 추론으로 처리함으로써 중복 호출을 최소화한다.

## Ⅱ. 연구 배경 및 관련 연구

### 2.1 LLM 기반 실시간 대화형 서비스

LLM은 수십억 개 이상의 파라미터를 기반으로 자연어를 이해하고 생성하며, 이 과정은 상당한 연산 자원을 요구한다[2]. LLM을 사용하는 대표적인 서비스는 API를 통해 입출력 토큰 수에 따른 종량제 방식으로 비용이 부과된다. 이에 따라 LLM 호출 횟수는 서비스 운영 비용과 직결되는 핵심 요소이다.

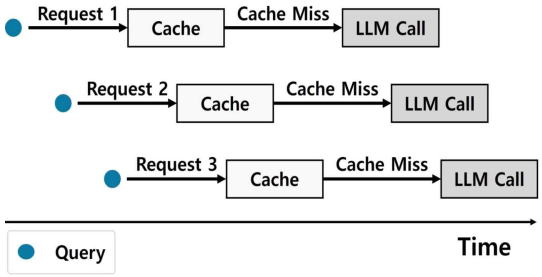


**Fig. 1** Redundant LLM calls under consecutive similar request

### 2.2 시맨틱 캐싱

LLM 서비스의 효율성을 높이기 위해 질의 응답을 캐시에 저장하고 재사용하는 방법이 연구되어 왔다. 전통적인 캐싱은 문자열의 정확한 일치를 기준으로 동작하지만, 동일 의도를 다양하게 표현하는 자연어 특성상 캐시 적중률이 낮다. GPTCache[5], GPT Semantic Cache[6], MeanCache[4] 등의 시맨틱 캐싱 연구는 질의를 임베딩 벡터로 변환하고 유사도 검색을 통해 과거 응답을 재사용함으로써 응답 지연과 비용을 개선하였다. 그러나 이들은 개별 요청 단위의 캐시 조회를 전제하여, 다수의 유사 질의가 연속으로 유입될 때는 충분히 대응하지 못한다. 또한 캐시 조회 이전에 유사 질의를 사전 그룹화하여 중복 호출을 방지하는 접근은 기존에 다루어지지 않았다.

### 2.3 연속 유사 요청 처리의 한계

그림. 1은 시맨틱 캐싱이 연속 유사 요청 처리 과정에서 구조적인 사각지대를 나타낸다. 첫 번째 요청이 캐시 미스로 판정되어 LLM을 호출하고 응답을 생성하는 동안, 의미가 유사한 요청들이 유입되면 이들 역시 캐시에서 일치하는 항목을 찾지 못해 각각 LLM을 호출하게 된다. 즉, 시맨틱 캐싱은 이미 처리된 질의에 대해서는 효과적이지만, 연속 유사 요청 상황에서는 중복 연산을 방지하지 못하는 한계를 가진다.

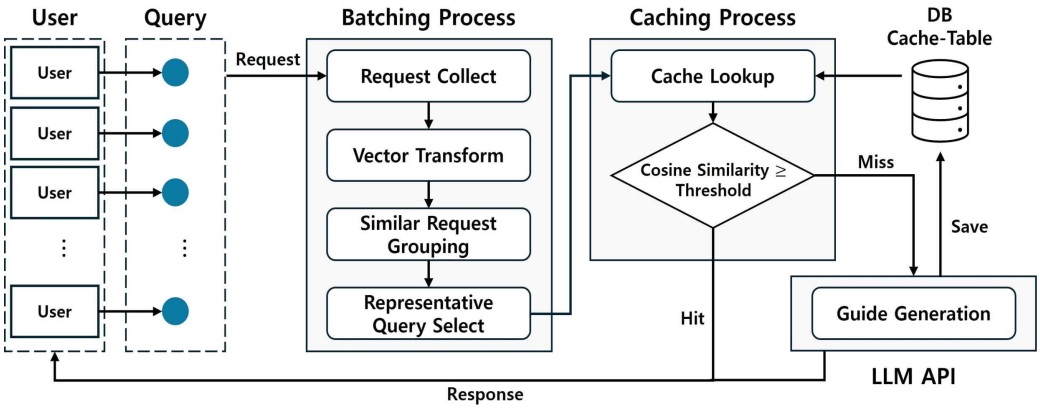


**Fig. 2** Overview of an LLM call optimization method using semantic dynamic batching and caching

## Ⅲ. 의미 기반 동적 배칭

본 장에서는 제안하는 의미 기반 동적 배칭 시스템의 구조와 임베딩 및 유사도 측정 방식을 설명한다.

### 3.1 의미 기반 동적 배칭 개요

연속 유사 요청으로 인한 중복 처리를 완화하기 위해서는 캐싱 이전 단계에서의 요청 병합 메커니즘이 필요하다. 이를 위해 본 연구는 의미 기반 동적 배칭(Semantic Dynamic Batching)을 제안한다.

의미 기반 동적 배칭은 LLM 호출 이전에 유사한 질의를 사전 병합하여 질의 총량 자체를 감소시키는 전처리 기법이다. 이는 처리 속도 향상이 아니라, 중복 연산 제거를 통한 비용 절감을 주요 목표로 한다.

### 3.2 시스템 구조

그림. 2는 제안하는 시스템의 전체 아키텍처를 나타낸다. 제안 시스템은 배칭 프로세스(Batching Process)와 캐싱 프로세스(Caching Process)의 두 단계로 구성되며, 핵심 설계 원칙은 캐시 조회 이전에 배칭을 수행하는 것이다. 배칭 단계를 캐싱 이전에 수행하여 다수의 질의를 의미적 그룹으로 압축한 후, 그룹당 대표 질의에 대해서만 캐시 처리를 수행한다.

배칭 프로세스는 배치 윈도우 동안 수집된 질의들을 임베딩 벡터로 변환하고, 유사도 기반으로 그룹화하여 각 그룹의 대표 질의를 선정한다. 이 과정을 통해 유사한 질의들이 하나의 그룹으로 병합되어, 전체 질의 수보다 적은 수의 대표 질의만 캐싱의 대상이 된다. 배칭 프로세스의 상세 동작은 3.4절에서 기술한다.

캐싱 프로세스는 대표 질의에 대해서만 캐시 조회를 수행한다(Cache Lookup). 대표 질의의 벡터와 캐시 테이블에 저장된 기존 질의 간 코사인 유사도를 계산하여, 임계값 이상인 항목이 존재하면 캐시 적중으로 판정하고 저장된 응답을 즉시 반환한다. 캐시 미적중 시에는 LLM API를 호출하여 새 응답을 생성하며, 생성된 응답은 대표 질의의 임베딩 벡터와 함께 캐시 테이블에 저장된다. 최종적으로 응답은 대표 질의뿐 아니라 동일 그룹 내 모든 질의에 동일하게 반환된다.

### 3.3 임베딩 및 유사도 계산

본 시스템의 배칭과 캐싱은 모두 임베딩 기반 유사도 계산을 기반으로 한다. 사용자 질의는 bge-m3 모델[8]을 통해 1,024차원 벡터로 변환된다. 두 질의 간의 유사성은 코사인 유사도[7]로 측정하며, 본 시스템에서는 배칭과 캐싱에 동일하게 0.85의 임계값을 적용하였다. 임계값 선정에 대한 분석은 4.4절에서 후술한다.

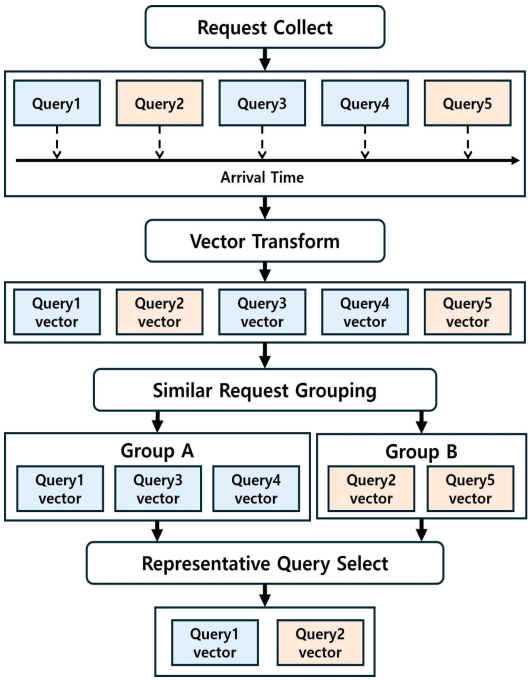


**Fig. 3** Semantic dynamic batching process

### 3.4 의미 기반 동적 배칭 프로세스

그림. 3은 의미 기반 동적 배칭의 처리 과정을 나타낸다. 배칭 프로세스는 네 단계로 구성된다. 첫째, 배치 윈도우(Batch Window) 동안 유입되는 사용자 요청들을 수집한다(Request Collect). 배치 윈도우는 요청을 그룹화하기 위해 설정하는 수집 대기 시간으로, 본 연구에서는 사용자 경험을 저해하지 않는 범위 내에서 200ms로 설정하였다. 둘째, 수집된 각 질의를 임베딩 모델을 통해 고차원 벡터로 변환한다(Vector Transform). 셋째, 변환된 벡터들을 유사도 기반으로 그룹화한다(Similar Request Grouping). 모든 질의 쌍에 대해 코사인 유사도를 계산하여, 임계값(0.85) 이상인 질의들을 동일 그룹으로 묶는다. 이 과정을 통해 의미적으로 유사한 질의들이 동일 그룹으로 분류된다. 넷째, 각 그룹에서의 대표 질의를 선정한다(Representative Query Select). 응답 지연 최소화를 위해 가장 먼저 도착한 질의를 대표 질의로 선정한다. 선정된 대표 질의만이 캐시 조회 및 LLM 호출 대상이 되며, 동일 그룹 내의 다른 질의들은 대표 질의의 응답을 공유함으로써 중복 연산을 방지한다.

## Ⅳ. 실 험

### 4.1 실험 환경 및 실험 데이터

임베딩 모델은 다국어를 지원하는 bge-m3[8]를 사용하였으며, 각 질의는 1,024차원 벡터로 변환된다. LLM은 vLLM[9] 기반 추론 서버에서 Qwen2.5-3B-Instruct[10] 모델을 사용하였다. 유사도 임계값은 0.85로 설정하였으며, 의미 기반 배칭을 위한 배치 윈도우는 200ms로 구성하였다.

테스트 질의는 비트코인 지갑 앱의 주요 기능을 기준으로 송금, 잔액 확인, 설정, 거래 내역의 4개 의도 그룹으로 구성하였다. 각 그룹당 서로 다르게 표현한 질의 5개씩, 총 20개의 유사 질의를 사용하였다.

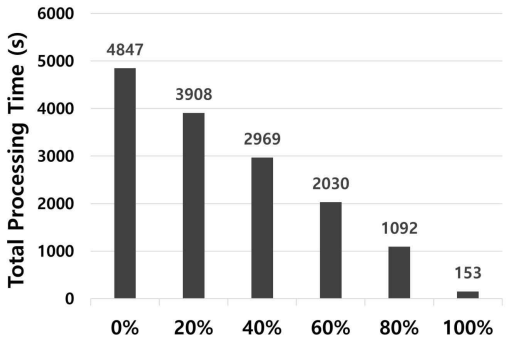


**Fig. 4** Response time with respect to cache hit rate

### 4.2 캐시 적중률에 따른 응답 시간

캐시 적중률이 응답 시간에 미치는 영향을 측정하였다. 그림. 4는 캐시 적중률 변화에 따른 평균 응답 시간을 나타낸다. 캐시 적중률 0%일 때 평균 응답 시간은 4,847ms이며, 100%일 때는 153ms로 약 32배 단축되었다. 캐시 적중 시 LLM 호출이 생략되어 응답 시간이 크게 단축됨을 확인하였다. 이 결과는 시맨틱 캐싱이 LLM 호출 비용 절감에 효과적임을 보여준다.

### 4.3 배칭 적용에 따른 추론 효율성 및 응답 지연 분석

20개 요청에 대해 배칭 유무에 따른 LLM 호출 횟수와 처리 시간을 측정하였다. 그림. 5는 캐싱 단독 적용과 캐싱과 배칭을 결합한 총 처리 시간을 비교한 결과이다. 캐싱만 적용한 경우 총 처리 시간은 13.478초이며, 배칭을 함께 적용한 경우 7.626초로 약 43% 단축되었다. 그림. 6은 LLM 호출 횟수를 비교한 결과이다. 배

칭 적용 시 LLM 호출 횟수가 20회에서 4회로 80% 감소하였다. 이는 20개 요청이 4개 그룹으로 그룹화되어 대표 질의에 대해서만 LLM을 호출하였기 때문이다. 배칭 적용 시 배치 윈도우(200ms) 동안의 대기 시간과 그룹화 처리 시간이 추가되지만, LLM 호출 횟수가 감소함에 따라 전체 처리 시간은 오히려 단축되었다.

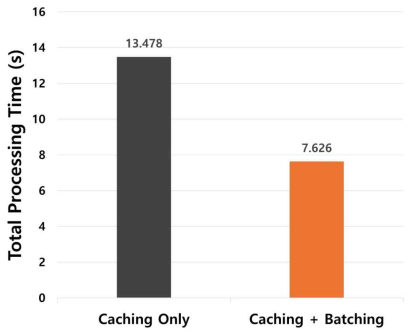


**Fig. 5** Comparison of processing time with and without batching

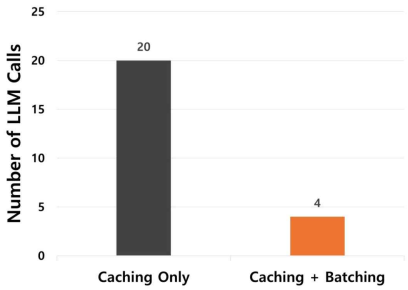


**Fig. 6** Comparison of the number of LLM calls with and without batching

이는 LLM 추론 시간이 배칭 오버헤드보다 훨씬 크기 때문이다. 한편, 배치 윈도우로 인한 200ms의 추가 지연은 1초 이내 응답 기준[11]을 충족하여 사용자 체감에 미치는 영향이 제한적이다. 결과적으로 배칭 적용은 LLM 호출 비용 절감과 처리 시간을 단축함을 보여준다.

### 4.4 유사도 임계값 분석

유사도 임계값 설정에는 트레이드오프가 존재한다. 임계값이 낮으면 의미가 다른 질의도 유사하다고 판단되어 부적절한 응답을 제공한다. 반대로 임계값이 높으면 의미가 유사한 질의도 다르다고 판단하여 캐시 미스가 증가한다. 본 시스템에서는 부적절한 응답 반환이 캐시 미스보다 치명적이라고 판단하였다. 캐시 미스는 응답 지연과 비용 증가를 유발하지만 응답의 정확성은 보장된다. 반면 부적절한 응답은 서비스 자체의 신뢰를 훼손한다. 이러한 이유로 정밀도에 가중치를 두는 F0.5 Score를 핵심 평가지표로 선정하였다.

그림. 7은 유사도 임계값 변화에 따른 정밀도, 재현율, F0.5 Score를 나타낸다. 임계값이 낮을수록 재현율은 높아지나 부적절한 응답의 위험이 증가하여 정밀도가 낮아진다. 반대로 임계값이 높을수록 정밀도는 높아지나 재현율이 감소한다. 실험 결과, 임계값 0.85에서 F0.5 Score가 0.9로 최댓값을 기록하였다. 이때 정밀도는 1.0으로 부적절한 응답이 완전히 제거되었으며, 캐시 효율도 적절히 유지되었다. 임계값이 0.85 미만일 경우 부적절한 응답이 발생할 위험이 있고, 0.85 초과일 경우 캐시 효율이 급격히 저하된다. 따라서 본 시스템에서는 0.85를 최적 임계값으로 선정하였다.

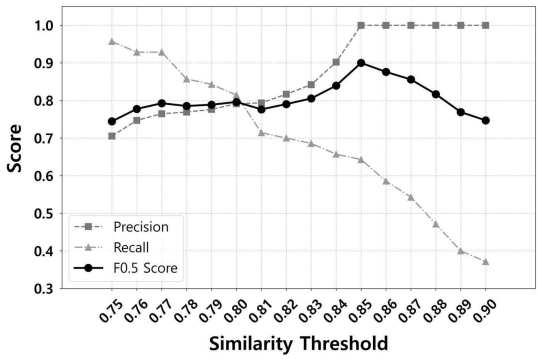


**Fig. 7** Precision, Recall, and F0.5 Score with respect to similarity thresholds

## Ⅴ. 결 론

본 논문은 LLM 기반 실시간 대화형 서비스에서 반복적으로 발생하는 의미적 유사 질의로 인해 초래되는 중복 LLM 호출 문제를 해결하기 위해, 의미 기반 동적 배칭과 시맨틱 캐싱을 결합한 LLM 호출 최적화 기법을 제안하였다. 제안 시스템은 배칭 프로세스와 캐싱 프로세스를 분리하고, 캐시 조회 이전 단계에서 bge-m3 임베딩과 코사인 유사도에 기반한 질의 그룹화를 수행함으로써 연속 유사 요청 구간에서 발생하는

구조적 사각지대를 해소한다. 그 결과, 비트코인 지갑 앱 시나리오를 이용한 실험에서 20개 요청이 4개의 의미 그룹으로 압축되면서 LLM 호출 횟수가 80% 감소하고, 캐싱만 적용한 경우와 비교해 총 처리 시간이 약 43% 단축되는 등 호출 비용과 지연 시간 측면에서 유의미한 성능 향상을 확인하였다. 다만, 본 연구의 실험은 초기 동작 가능성과 효과를 확인하는 데 초점을 맞추었기에, 4개 의도와 각 5개 표현으로 구성된 20개 질의만을 활용하여 실험 규모가 제한적이다. 또한, 긴 질의나 의도 혼합 질의 등 실제 서비스 환경의 다양한 조건은 반영되지 못하였다는 한계가 존재한다.

향후 연구에서는 고객센터, 교육 챗봇 등 다양한 도메인과 수백~수천 건 규모의 동시 요청 환경으로 실험을 확장하여, 제안 기법의 일반화 가능성과 확장성을 검증하고자 한다.

## ACKNOWLEDGEMENTS

This research was financially supported by Hansung University.

## REFERENCES

[1] S. K. Dam, C. S. Hong, Y. Qiao, and C. Zhang, “A complete survey on LLM-based AI chatbots,” *arXiv preprint arXiv: 2406.16937*, Jun. 2024. DOI: 10.48550/ arXiv.2406.16937.

[2] Z. Wan, X. Wang, C. Liu, S. Alam, Y. Zheng, J. Liu, Z. Qu, S. Yan, Y. Zhu, Q. Zhang, et al., “Efficient large language models: A survey,” *arXiv preprint arXiv: 2312.03863*, Dec. 2023. DOI: 10.48550/arXiv.2312.03863.

[3] Z. Zhou, X. Ning, K. Hong, T. Fu, J. Xu, S. Li, Y. Lou, L. Wang, Z. Yuan, X. Li, et al., “A survey on efficient inference for large language models,” *arXiv preprint arXiv: 2404.14294*, Apr. 2024. DOI: 10.48550/arXiv.2404.14294.

[4] W. Gill, M. Elidrisi, P. Kalapatapu, A. Ahmed, A. Anwar, and M. A. Gulzar, “MeanCache: User-centric semantic caching for large language model based web services,” *arXiv preprint arXiv: 2403.02694*, Mar. 2024. DOI: 10.48550/arXiv.2403.02694.

[5] F. Bang, “GPTCache: An open-source semantic cache for LLM applications enabling faster answers and cost savings,” in *Proceeding of the 3rd Workshop for Natural Language Processing Open Source Software (NLP-OSS 2023)*, Singapore: SG, pp. 212-218, 2023. DOI: 10.18653/ v1/2023.nlposs-1.24.

[6] S. Regmi and B. Pun, “GPT semantic cache: Reducing LLM costs and latency via semantic embedding caching,” *arXiv preprint arXiv: 2411.05276*, Nov. 2024. DOI: 10.48550/arXiv.2411.05276.

[7] H. Steck, C. Ekanadham, and N. Kallus, “Is cosine-similarity of embeddings really about similarity?,” *arXiv preprint arXiv: 2403.05440*, Mar. 2024. DOI: 10.48550/ arXiv.2403.05440.

[8] J. Chen, S. Xiao, P. Zhang, K. Luo, D. Lian, and Z. Liu, “M3-Embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation,” *arXiv preprint arXiv: 2402.03216*, Feb. 2024. DOI: 10.48550/arXiv.2402.03216.

[9] W. Kwon, Z. Li, S. Zhuang, Y. Sheng, L. Zheng, C. H. Yu, J. E. Gonzalez, H. Zhang, and I. Stoica, “Efficient memory management for large language model serving with PagedAttention,” in *Proceeding of the 29th ACM Symposium on Operating Systems Principles (SOSP 2023)*, Koblenz: DE, pp. 611-626, 2023. DOI: 10.1145/3600006.3613165.

[10] A. Yang, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Li, D. Liu, F. Huang, H. Wei, et al., “Qwen2.5 technical report,” *arXiv preprint arXiv: 2412.15115*, Dec. 2024. DOI: 10.48550/arXiv.2412.15115.

[11] J. Nielsen, *Usability Engineering*, 1st ed. New York, NY: Academic Press, 1993.

**김소룡(So-Ryong Kim)**

2020 – 2026년 한성대학교 컴퓨터공학부 졸업
2026년 - 현재 한성대학교 AI응용학과 석사 과정
※ 관심분야 : 딥러닝, 시스템 최적화, 병렬 컴퓨팅

**김명선(Myung-Sun Kim)**

2000년 - 2002년 LG전자 전송연구소 주임연구원
2002년 - 2011년 삼성전자 DMC 연구소 책임연구원
2016년 서울대학교 전기컴퓨터공학부 박사 졸업.
2016년 - 2019년 삼성전자 SR연구소 수석연구원
2019 - 현재 한성대학교 AI응용학과 부교수
※ 관심분야 : NPU(인공지능 프로세서), Linux kernel, HW/SW Co-design, 객체인식, 딥러닝 가속 및 병렬 컴퓨팅