인공지능 모델 및 결과서

1. 모델 선정 및 선정 이유

💡 단일 모델을 사용하는 대신, 특정 작업에 특화된 여러 개의 모델을 나누어 배치했습니다.

구분	모델명	주요 역할 (Task)	선정 사유 및 기대 효과
Main LLM	`gpt-4.1-mini`	전체 맥락 요약 및 후처리	복잡한 추론 및 대화 흐름 파악 시 높은 정확도 보장
Domain SLM	`Llama-3-Kor-BCCard-8B`	금융 도메인 지식 추출	BC카드 특화 데이터 기반으로 금융 약관 및 내부 규정 이해도 우수
Persona SLM	`EXAONE-3.5-2.4B`	상담원 교육 대상 대화 페르소나 구현	한국어 특유의 뉘앙스 및 격식체 구현을 통한 상담 몰입감 증대

구분	모델명	주요 역할 (Task)	선정 사유 및 기대 효과
Embedding	`text-embedding-3-small`	RAG 벡터 검색 최적화	다국어 처리 강점 및 검색 정확도 대비 낮은 인프라 비용

해당 전략의 채택 이유

- **비용 최적화**
  - 고비용 LLM과 저비용 sLLM을 혼합 구성하여, 단일 LLM 운영 대비 API 호출 비용 절감 예상
- **응답 속도 개선**
  - 경량화된 sLLM 배치를 통해 단순 질의 및 도메인 지식 조회 시 시스템 응답 시간 단축 예상
- **정확도 향상**
  - 범용 AI가 놓치기 쉬운 카드사 특화 금융 용어 및 내부 업무 경로에 대한 응답 신뢰도 확보

2. 모델 파라미터 설정 및 주요 시스템 파라미터 최적화

> 💡 상담 응답의 일관성, 지연 시간 최소화, 환각 방지를 목표로 모델 파라미터를 보수적으로 설정했습니다.

주요 목적과 적용 내역

2-1. LLM 추론 제어 및 비용 최적화

- **Hallucination 제어**
  - `gpt-4.1-mini` , `temperature = 0.0` : 응답 변동성을 줄여 동일 질의에 대한 결과 일관성 확보
- **Latency & Cost 최적화**
  - `llm_card_top_n = 2` : 모든 검색 결과가 아닌 상위 핵심 2개만 요약 생성에 사용
  - `450 chars` : LLM 문서 길이를 제한하여 정보 과부하로 인한 답변 품질 저하 방지 및 추론 속도 향상

2-2. 검색 및 캐시 시스템 설정

- **검색 정밀도**
  - `top_k = 5` : 초기 검색 단계에서 5개의 후보군을 확보 후, 랭킹 알고리즘(RRF)을 통해 최적의 정보를 선별함
- **응답 가속화**
  - Redis 캐시 전략: 동일/유사 질의에 대해 TTL 120초의 캐시를 적용. 캐시 적중(Hit) 시 LLM 추론 과정을 생략하여 1초 내외의 초저지연 응답 가능
  - Fail-open 구조: 캐시 서버 장애 시 즉시 DB/LLM 직접 조회로 전환하여 서비스 중단 없는 연속성(High Availability) 보장

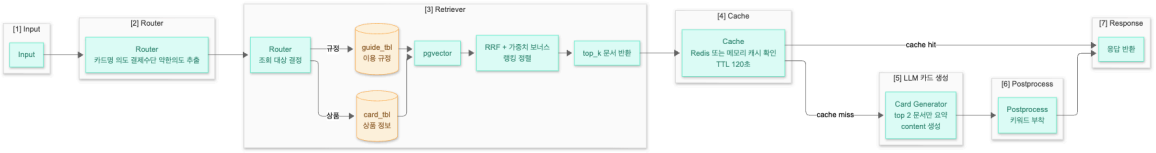
3. 전체 시스템 구조

💡 본 시스템은 고객의 질의 의도를 정밀하게 분석하고, 카드사 도메인 지식(Knowledge Base)을 결합하여 최적의 응답을 생성하는 Advanced RAG(Retrieval-Augmented Generation) 구조를 채택하고 있습니다.

RAG 파이프라인 구조 및 흐름 설명

단계	프로세스	주요 기술 및 수행 내용
1. 입력	Input & STT	• 상담원/고객의 음성 실시간 텍스트화 • 상담원이 입력한 검색 텍스트
		도메인 어휘 사전: 카드사 특화 전문 용어를 보정하여 핵심 키워드(Keywords) 추출 성능 최적화
2. 분석	Routing	Intent 분석: 추출된 텍스트에서 카드명, 혜택, 결제수단 등 Entity와 상담 의도(Intent) 파악
		쿼리 최적화: 분석된 의도에 따라 질문을 상품 정보 또는 이용 규정 DB 중 최적의 저장소로 라우팅
3. 검색 (Hybrid Retrieval)	Hybrid Retrieval	Search: `pgvector` 기반의 Vector 검색
	RRF & Re-ranking	RRF & Re-ranking: 두 검색 결과를 RRF(Reciprocal Rank Fusion) 알고리즘으로 결합하고 비즈니스 가중치 보너스 적용
4. 생성 (Generation)	Cache-First Layer	Card Cache 확인: Redis 기반 캐시를 우선 조회하여 Hit 시 즉시 반환(Latency 절감)
		Cache Miss 시, 상위 2개 문서를 GPT-4.1-mini에 주입하여 요약 생성
5. 정제 (Post-processing)	Post-processing	Validation: 생성된 JSON 객체의 정합성 검증 및 빈 필드 제거
		UI 최적화: 문단 분리 및 키워드 해시태그화(#) 작업을 거쳐 상담 요약 리포트로 최종 반환

시스템 구성도



4. 프롬프트 엔지니어링

💡 상담 데이터의 정합성을 확보하고 LLM 추론 비용을 최소화하기 위해 지시사항 고도화 및 출력 구조 단순화 전략을 적용하였습니다.

항목	주요 내용	기대 효과
구조적 제어	`response_format={"type":"json_object"}` 적용	출력 파싱 오류 원천 차단 및 시스템 안정성 확보
정보 제약	문서 기반 응답 (Groundedness) 강제	외부 지식 개입을 차단하여 환각 (Hallucination) 현상 방지
토큰 최적화	`llm_card_top_n = 2` 설정	불필요한 카드 생성을 제한하여 비용 절감 및 생성 속도 향상
입력 제한	`MAX_CARD_DOC_CHARS = 450`	컨텍스트 윈도우 최적화로 추론 효율 극대화

4.1. 프롬프트 예시 및 개선 과정

- **최초 프롬프트**

```
f"""다음은 카드 상담용 문서입니다. 사용자 질문과 문서를 참고해 카드 상세 정보를 생성하세요.

### 지시 사항
1. 정확한 정보가 없으면 합리적인 상담 시나리오를 간단히 구성해도 됩니다.
2. 반드시 JSON 객체만 반환하세요. 추가 텍스트는 금지합니다.
3. 카드 수는 총 {doc_count}개이며, 제공된 문서의 순서와 동일하게 cards 배열을 구
```

```
성하세요.
4. 모든 카드에 모든 필드를 채우되, 알 수 없으면 빈 문자열/빈 배열로 채우세요.
5. guidanceScript는 문서에서 그대로 발췌한 문장만 사용하세요.
6. 문서에 없는 내용은 절대 추가하지 마세요.

### 사용자 질문
{query}

### 상담 문서 내용
{joined_docs}

### 출력 형식 (JSON Schema)
{{
  "cards": [
    {{
      "id": "문서 id 그대로",
      "title": "문서 title 그대로",
      "keywords": ["#키워드1", "#키워드2"],
      "content": "사용자에게 보여줄 1~2문장 요약",
      "systemPath": "업무 경로(없으면 빈 문자열)",
      "requiredChecks": ["필수 확인 사항"],
      "exceptions": ["예외 사항"],
      "regulation": "관련 규정/약관(없으면 빈 문자열)",
      "time": "처리 시간(없으면 빈 문자열)",
      "note": "추가 메모(없으면 빈 문자열)"
    }}
  ],
  "guidanceScript": "상담원이 읽을 간단 스크립트(없으면 빈 문자열)"
}}
"""
```

- **개선 프롬프트**

```
f"""다음은 카드 상담용 문서입니다. 사용자 질문과 문서 내용을 참고해 카드 요약(content)만 생성하세요.

 ### 지시 사항
```

```
  1. 반드시 아래 제공된 JSON 객체 형식만 반환하세요. 추가 텍스트는 금지합니다.
  2. 카드 수는 총 {doc_count}개이며, 문서의 순서와 동일하게 cards 배열을 구성하세요.
  3. 각 요약은 1~2문장으로 작성하며, 문서에 없는 내용은 절대 포함하지 마세요.
  4. content 외의 필드는 출력하지 마세요.

### 사용자 질문
  {query}

  ### 상담 문서 내용
  {joined}

  ### 출력 형식 (JSON Schema)
  {{
  "cards": [
     {{ "content": "카드 요약 내용 1-2문장" }}
  ]
  }}"""
```

- **개선 방향**: 불필요한 JSON 필드(id, title, systemPath 등)를 고정 데이터로 처리하고, LLM은 가변 데이터인 요약문 생성에만 집중하도록 지시문을 간소화

- **프롬프트 변경 전후 성능 변화**

```
# 원본 로그

# 변경 전
[rag] route=0.3ms retrieve=1742.3ms cards=13940.4ms post=0.6ms tot
al=15683.6ms

# 변경 후
[rag] route=0.4ms retrieve=1164.7ms cards=10542.4ms post=0.6ms tot
al=11708.2ms docs=4 route=card_usage
```

구분	전체 레이턴시 (Total Latency)	**생성 소요 시간**
개선 전	15.6s	13.9s
개선 후	**11.7s**	**10.5s**

개선 결과	약 25% 속도 개선	추론 부하 감소 및 응답 속도 최적화 성공

5. 성능 평가 지표

지표명	측정 목적	목표 및 기준
Recall@3	금융/카드 핵심 키워드 인식률 검증	Top-3 결과 내 핵심 정보 포함 여부 (목표: 0.9 이상)
Faithfulness	검색 문서 근거 기반 답변 생성 여부	환각(Hallucination) 발생 여부 정밀 검증 (RAGAS 활용)
Latency	실시간 상담 가동 가능성 판단	평균 응답 속도 3초 이내 (Cache Hit 시 1초대 목표)
Macro F1	상담 카테고리 분류의 정확성	카드 발급/혜택 등 분류 결과의 균형적 정확도 측정

6. 테스트 결과 및 종합 분석

6-1. 목적 및 환경

항목	내용
테스트 목적	실시간 상담 지원 RAG 시스템의 응답 정확성, 신뢰성, 지연 시간 검증
상세 목적 및 시나리오	카드 발급·재발급·혜택·분실 등 실제 상담 빈도가 높은 질의를 기반으로 STT → RAG → 카드 생성 파이프라인 검증
테스트 환경	macOS / Python 3.11 / PostgreSQL + pgvector / Redis Cache / OpenAI GPT-4.1-mini
입력 데이터	카드 관련 질의 12건 (MVP 검증용 소규모 샘플(12건) 기반, 향후 질의셋을 확대하여 재평가 예정)
기대 출력	{ "id": "narasarang_faq_004", "title": "나라사랑카드를 신규로 발급받으려면 어떻게 하나요?", "content": "나라사랑카드 신규 발급은 병역판정검사(징병검사)자를 대상으로 전국 병역판정검사(징병검사)장내의 IBK기업은행, 테디은행의 나라사랑카드 발급소에서 신규 발급이 가능합니다. 나라사랑카드 시행일(2007.1.29.) 이전에 징병검사를 받은 분은 신규발급이 불가하며, 육군훈련소로 입영할 경우 훈련소에서 발급받으실 수 있습니다. 병역판정검사(징병검사) 시 나라사랑카드는 IBK기업은행, 테디은행의 나라사랑카드 중 1개 은행을 선택하여 발급받을 수 있고, 추후 나라사랑카드 추가 발급은 은행 영업점 등에서 신청할 수 있습니다. 나라사랑카드는

항목	내용

	본인 희망 시에만 발급하고 있으며, 강제 발급 사항이 아닙니다. 나라사랑카드 시행일인 2007.1.29. 이전에 병역판정검사(징병검사)를 받은 의무자는 신규발급이 불가하고, 나라사랑카드는 병역의 의무를 수행하기 위하여 병역 판정을 받는 만 19세 남성인 병역판정검사(징병검사)자를 대상으로 발급됨에 따라, 병역의 의무가 없는 여성(여군 포함)은 발급받을 수 없습니다. 2007~2015년 신한은행 나라사랑카드를 발급받은 의무자는 IBK기업은행, 테디은행 나라사랑카드를 추가로 발급 받아 소지/사용 할수 있습니다.",
	"text": "나라사랑카드를 신규로 발급받으려면 어떻게 하나요?\n나라사랑카드 신규 발급은 병역판정검사(징병검사)자를 대상으로 전국 병역판정검사(징병검사)장 내의 IBK기업은행, 테디은행의 나라사랑카드 발급소에서 신규 발급이 가능합니다. 나라사랑카드 시행일(2007.1.29.) 이전에 징병검사를 받은 분은 신규발급이 불가하며, 육군훈련소로 입영할 경우 훈련소에서 발급받으실 수 있습니다. 병역판정검사(징병검사) 시 나라사랑카드는 IBK기업은행, 테디은행의 나라사랑카드 중 1개 은행을 선택하여 발급받을 수 있고, 추후 나라사랑카드 추가 발급은 은행 영업점 등에서 신청할 수 있습니다. 나라사랑카드는 본인 희망 시에만 발급하고 있으며, 강제 발급 사항이 아닙니다. 나라사랑카드 시행일인 2007.1.29. 이전에 병역판정검사(징병검사)를 받은 의무자는 신규발급이 불가하고, 나라사랑카드는 병역의 의무를 수행하기 위하여 병역 판정을 받는 만 19세 남성인 병역판정검사(징병검사)자를 대상으로 발급됨에 따라, 병역의 의무가 없는 여성(여군 포함)은 발급받을 수 없습니다. 2007~2015년 신한은행 나라사랑카드를 발급받은 의무자는 IBK기업은행, 테디은행 나라사랑카드를 추가로 발급 받아 소지/사용 할수 있습니다.",
	"metadata": {
	"card_name": "나라사랑카드",
	"category": "발급/신청"
	}
	},

실제 출력	{ "id": "card_31", "title": "나라사랑카드를 신규로 발급받으려면 어떻게 하나요?", "keywords": ["#나라사랑카드", "#발급"], "content": "나라사랑카드는 병역의 의무가 있는 대상자에게 발급되며, 병역의 의무가 없는 여성(여군 포함)은 발급받을 수 없습니다. 2007~2015년 신한은행 나라사랑카드를 발급받은 의무자는 IBK기업은행, KB국민은행 나라사랑카드를 추가로 발급받을 수 있습니다.", "requiredChecks": ["병역의무 대상자 여부 확인"], "exceptions": ["병역의 의무가 없는 여성(여군 포함)는 발급 불가"

항목	내용
	], "detailContent": "대상으로 발급됨에 따라, 병역의 의무가 없는 여성(여군 포함)은 발급받을 수 없습니다. 2007~2015년 신한은행 나라사랑카드를 발급받은 의무자는 IBK기업은행, KB국민은행 나라사랑카드를 추가로 발급 받아 소지/사용할수 있습니다.", }
응답 소요 시간	평균 2.6초 (miss), 캐시 hit 시 1초대
분석 및 개선점	LLM 카드 생성 단계가 주요 지연 원인 → 캐시 및 프롬프트 축소로 개선

항목	주요 성과 및 데이터	비고	목표
정확도	Recall@3: 0.9 Macro F1: 0.9	검색 및 의도 분류의 완벽한 정합성 확인	0.9 이상
신뢰성	Faithfulness: 0.94 RLHF: 4.0	근거 기반 답변 생성으로 환각 현상 유의미하게 억제	0.9 4.0 이상
응답성	평균 2.6초 (Cache Hit 시 1초 내외)	실시간 상담 프로세스 저해 없는 Latency 달성	2초대

7. 결론 및 향후 개선 방향

- **결과 요약**
  - 평균 2.6초 응답 속도로 실시간 상담 지원에 적합
  - Recall@3, Macro F1 = 0.9로 검색 및 분류 안정성 확보
  - Faithfulness 0.94, RLHF 4.0로 신뢰성 및 사용자 만족도 검증 완료
  - Redis 캐시 적용 시 hit 기준 1초대 응답 가능
- **한계 및 개선점**
  - 현재 시스템은 모든 질의에 대해 동일한 검색·생성 파이프라인을 거치기 때문에, 단순 조회 질문에서도 불필요한 연산이 발생하는 한계가 있다.
  - 정보 생성 단계에서 LLM 의존도가 높아, 트래픽 증가 시 응답 지연 가능성이 존재한다.
- **향후 개발 계획 및 방향**

- 실시간 상담 흐름을 반영하기 위해 이전 질의 맥락을 활용한 멀티턴 상담 지원 기능을 추가할 계획이다.
- 반복적으로 사용되는 카드 정보에 대해 사전 요약 및 캐시 기반 응답 구조를 도입하여 응답 속도와 안정성을 개선할 예정이다.
- 향후에는 상담원 교육용 시뮬레이션 기능을 확장하여, 가상 고객과의 대화 및 상담 품질 피드백까지 지원하는 방향으로 발전시키고자 한다.