테스트 계획 및 결과 보고서

1. 테스트 개요 및 목적

> 💡 본 테스트는 실시간 카드 상담 환경에서 STT 인식 정확도와 RAG 기반 답변의 신뢰성/신속성을 검증하기 위해 수행되었습니다. 특히 초기 모델의 긴 지연 시간(Latency)을 해결하기 위한 단계별 최적화 과정을 트래킹하여 실사용 가능성을 확증하는 데 목적이 있습니다.

2. 테스트 시나리오 및 방법

2.1 RAG 테스트 시나리오

실제 상담 상황을 가정한 대표 질의 12개를 선정해 반복 실행하였다.

질의 유형은 발급/재발급/혜택/분실/사용처 등으로 구성하였다.

테스트 처리 흐름 (반복 수행 단계)

1. 라우터를 통한 질의 분기 (`card_usage` / `card_info`)
2. Keyword = DB 문자열 검색
3. LLM을 통한 카드 정보 카드(cards) 생성
4. End-to-End 응답 시간 측정
5. 검색 정확도 및 생성 품질 평가(RAGAS/RLHF)

※ 성능 측정은 “실서비스 worst-case”를 가정하여 **cache=miss 기준**을 원칙으로 하되, 캐시 효과 확인을 위해 hit 로그도 별도로 기록하였다.

2.2 STT 성능 및 시간 확인 시나리오

카드 상담 실제 녹취 데이터를 기반으로 대표 발화 시나리오 5개를 선정하여 진행하였다.

테스트 처리 흐름

1. 상담 음성 파일 입력 (MP3)
2. OpenAI Whisper-1 모델을 통한 텍스트 변환
3. 텍스트 정규화

```
def normalize(text):
    transformation = jiwer.Compose([
        jiwer.ToLowerCase(),
        jiwer.RemovePunctuation(),
        jiwer.RemoveMultipleSpaces(),
        jiwer.Strip(),
    ])
    return transformation(text)
```

4. Ground Truth(실제 정답 대본)와 STT 결과 비교 분석 및 지연 시간 측정

```
def calculate_metrics(truth, hypothesis):
    norm_truth = normalize(truth)
    norm_hyp = normalize(hypothesis)
    if not norm_truth: return 0.0, 0.0

    wer = jiwer.wer(norm_truth, norm_hyp)
    cer = jiwer.cer(norm_truth, norm_hyp)
    return wer, cer

wer, cer = calculate_metrics(ground_truth, hypothesis)
latency = end_time - start_time
rtf = latency / audio_duration if audio_duration > 0 else 0
```

```
print(f"1. 평균 변환 시간 : {success_df['latency'].mean():.4f} 초")
print(f"2. p95 변환 시간  : {success_df['latency'].quantile(0.95):.4f} 초")
print(f"3. RTF (평균)     : {success_df['rtf'].mean():.4f}")
print(f"4. WER (단어 오류): {success_df['wer'].mean():.4f}")
print(f"5. CER (음절 오류): {success_df['cer'].mean():.4f}")
```

3. 테스트 환경

3-1. RAG 테스트 인프라 환경

시스템의 재현 가능성과 로컬 검증 성능을 확인하기 위해 아래 표준 개발 환경에서 테스트를 수행하였습니다.

- **OS/Runtime**: macOS / Python 3.11
- **Database**: PostgreSQL 16 + `pgvector` (Vector Search 엔진)
- **Cache Layer**: Redis (In-memory 캐시 및 Fail-over 검증용)
- **AI Model**: OpenAI `gpt-4.1-mini` (Main LLM), `text-embedding-3-small` (Embedding)
- 12개의 질의 테스트 셋


```
{[
 {
  "query": "나라사랑카드 재발급",
  "expect_route": "card_usage",
  "expect_should_route": true
 },
 {
  "query": "나라사랑카드란 무엇인가요",
  "expect_route": "card_info",
  "expect_should_route": true
 },
 {
  "query": "국민행복카드 분실",
  "expect_route": "card_usage",
  "expect_should_route": true
 },
 {
  "query": "민생회복 소비쿠폰 신청 방법",
  "expect_route": "card_usage",
  "expect_should_route": true
 },
 {
  "query": "K-패스 혜택 뭐 있어요?",
  "expect_route": "card_info",
  "expect_should_route": true
```

```
  },
  {
    "query": "애플페이 편의점",
    "expect_route": "card_usage",
    "expect_should_route": true
  },
  {
    "query": "네이버 카드 시간",
    "expect_route": "card_info",
    "expect_should_route": true
  },
  {
    "query": "연회비 반환 기준 알려줘",
    "expect_route": "card_usage",
    "expect_should_route": false
  },
  {
    "query": "카드 분실했어요",
    "expect_route": "card_usage",
    "expect_should_route": true
  },
  {
    "query": "쿠팡 와우 적립",
    "expect_route": "card_usage",
    "expect_should_route": true
  },
  {
    "query": "다둥이카드 발급 방법",
    "expect_route": "card_usage",
    "expect_should_route": true
  },
  {
    "query": "교통카드로 사용할 수 있나요",
    "expect_route": "card_usage",
    "expect_should_route": false
  }
]
}
```

- **기대 응답**

개요	내용
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 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)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 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"],"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/사용 할수 있습니다.",}

3-2. RAG 성능 최적화 트래킹

단순 구현에 그치지 않고, 로그 분석을 통한 4단계 최적화를 진행하여 응답 속도를 약 80% 이상 개선하였습니다.

최적화 단계	주요 조치 내용 (Engineering Action)	응답 시간(Total)	개선율
Step 1. 초기 모델	RAG 기본 파이프라인 구성 및 전체 필드 생성 지시	**15.68s**	-
Step 2. 프롬프트 경량화	JSON Schema 간소화 및 가변 데이터(content) 생성 집중	**11.71s**	25% ↓
Step 3. 파라미터 튜닝	Retrieval `top_k` 최적화 (4 → 2) 및 검색 로직 효율화	**7.81s**	33% ↓
Step 4. 캐시/구조 최적화	**Redis TTL 캐시** 도입 및 생성 병목 구간 제거	**2.60s**	**최종 83% ↓**

3.3 [Engineering Log] 단계별 최적화 적용 내역 및 RAG 성능 추적

본 섹션은 RAG 기반 상담 시스템의 응답 지연 문제를 해결하기 위해 수행한 단계별 최적화 과정을 실제 시스템 로그와 함께 정리한 것이다.

각 단계마다 문제 원인 → 적용한 조치(Action) → 성능 변화(Log)를 명확히 매칭하여 성능 개선 과정을 추적 가능하도록 구성하였다.

단계 1. [Baseline] 초기 프롬프트 구성 (최적화 전)

적용 내용

- LLM에게 카드 상세 정보의 모든 필드(id, title, systemPath, requiredChecks 등 약 10개 항목)를 한 번에 생성하도록 지시

문제점

- 입력 컨텍스트 및 출력 토큰 수 과다
- 실시간 상담 시스템 요구사항(3초 이내 응답)에 부적합한 지연 발생

로그

```
# [Before] 모든 필드 생성 시도
[rag] route=0.3ms retrieve=1742.3ms cards=13940.4ms post=0.6ms total=15683.6ms docs=4 route=card_usage
```

단계 2. [프롬프트 개선] 출력 포맷 개선

적용 내용

- LLM이 반드시 필요한 `content(요약문)` 생성에만 집중하도록 프롬프트 구조 축소
- `response_format={"type":"json_object"}` 적용으로 출력 안정성 확보

효과

- 출력 구조 오류 감소
- 요약 생성 시간 점진적 감소 (약 20~30%)

로그

```
# [After] 프롬프트 경량화 적용
[rag] route=1.1ms retrieve=1295.9ms cards=14223.1ms post=0.5ms total=15
520.6ms docs=4 route=card_usage
[rag] route=0.4ms retrieve=1582.1ms cards=10970.1ms post=0.5ms total=1
2553.1ms docs=4 route=card_usage
```

단계 3. LLM 입력 축소

적용 내용

- LLM 요약 문서 수 제한: llm_card_top_n = 2
- 문서 길이 제한: MAX_CARD_DOC_CHARS = 450

목적

- LLM 입력 토큰/출력 길이 축소로 카드 생성 지연 감소

효과

- 입력 토큰 수 감소
- 카드 생성 시간 약 **40% 이상 단축**

로그

```
# [Action] LLM + llm_card_top_n=4  ⇒  LLM + llm_card_top_n=2
[rag] route=0.3ms retrieve=1377.3ms cards=6436.0ms post=0.5ms total=7
814.1ms docs=4 route=card_usag
```

단계 4. [속도 가속화] Redis 캐시 레이어 도입

적용 내용

- 동일 질의에 대해 TTL 120초 Redis 캐시 적용
- 카드 생성 결과를 재사용하도록 설계

효과

- 캐시 적중 시 LLM 호출 완전 제거
- End-to-End 응답 시간 **1초대 달성**

로그

```
# [Action] TTL 캐시 적용 후 반복 질의
[rag] route=0.5ms retrieve=1373.8ms cards=0.1ms post=0.1ms total=1374.5
ms docs=4 route=card_usage cache=hit
```

단계 5. [안정성 확보] 에러 핸들링 및 Fail-open 설계

적용 내용

- LLM 응답 오류 시 1회 재시도(백오프) 후 실패 처리
- Redis 장애 발생 시 DB 기반 검색으로 자동 Fallback (Fail-open)

효과

- 캐시 서버 장애 상황에서도 서비스 정상 동작 확인
- 단일 장애 지점(SPOF) 제거

로그

```
# [Action] Redis ConnectionError 강제 발생 테스트
[rag] redis cache get failed: ConnectionError("Error connecting to localhos
t:6379...")
[rag] route=0.8ms retrieve=1843.9ms cards=1451.1ms post=0.1ms total=32
95.9ms docs=4 route=card_usage cache=miss
```

요약

단계	적용 내용 (Optimization Action)	소요 시간(Total)	개선 효과
Stage 1	초기 RAG 파이프라인 (프롬프트 최적화 전)	15.68s	-

Stage 2	프롬프트 경량화 및 응답 형식(JSON) 고정	11.71s	25% 단축
Stage 3	요약문 제공(`top_n` 4→2 조정)	7.81s	33% 단축
Stage 4	Redis 캐시 적용 및 병목 로직 제거	2.60s	최종 83% 개선

3-4. STT 테스트 인프라 환경

- **OS/Runtime**: Window / Python 3.11
- **STT 엔진:** OpenAI Whisper-1
- **언어:** 한국어 (`language="ko"`)
- **데이터셋:** 상담 문의 녹음 샘플 및 텍스트 데이터 5개
  https://aihub.or.kr/aihubdata/data/view.do?currMenu=115&topMenu=100&aihubDataSe=data&dataSetSn=100


```
[
 {
  "id": "S00000365_0002.wav",
  "folder": "S00000365",
  "audio_path": "C:\\backend\\tests\\test_dataset\\wav\\S00000365\\0002.wav",
  "truth": "네 안녕하세요. 결제 때문에 전화 드렸어요."
 },
 {
  "id": "S00000365_0003.wav",
  "folder": "S00000365",
  "audio_path": "C:\\backend\\tests\\test_dataset\\wav\\S00000365\\0003.wav",
  "truth": "n/ 어/ 분당 살 때"
 },
 {
  "id": "S00000365_0005.wav",
  "folder": "S00000365",
  "audio_path": "C:\\backend\\tests\\test_dataset\\wav\\S00000365\\0005.wav",
  "truth": "지금은 학습지 이용 안 하고 있거든요. 그리고 저희 서초로 이사도 했고요."
```

```
  }
]
```

4. 성능 평가 결과

💡 RAG의 검색 정확도, 생성 신뢰성 및 STT 인식 품질은 각각 Recall@K, RAGAS Faithfulness, WER/CER 등 정량 지표를 통해 측정하였으며, 해당 지표들은 단일 요청 단위의 수치 집계 결과로 로그 형태의 단계별 트래킹은 제공되지 않는다.
다만, 각 평가지표는 실제 서비스 시나리오를 반영한 테스트 입력을 기준으로 반복 실행되었으며,
시스템 전반의 성능 수준을 정량적으로 비교·검증하는 데 목적을 두고 활용하였습니다.

4.1 RAG

검색 정확도

지표	결과	목표치	달성 여부
Recall@3	0.9	≥ 0.9	O
Macro F1-Score	0.9	≥ 0.9	O

생성 신뢰성 및 품질

지표	결과	해석
RAGAS Faithfulness	0.9375	문서 기반 생성 신뢰성 매우 높음
RLHF 평균 점수	4.0 / 5.0	실사용 가능 수준

요약 품질

지표	결과	비고
ROUGE-L 평균	**0.17**	상담 스크립트 특성상 불필요한 수식어를 배제하고 핵심 키워드(Key-Entity) 위주로 답변하도록 튜닝되어 수치가 낮게 측정됨

4.2 STT

```
[
 {
  "file": "S00000365_0002.wav",
  "duration": 3.157,
  "latency": 1.439,
  "rtf": 0.456,
  "wer": 0.5,
  "cer": 0.091,
  "truth": "네 안녕하세요. 결제 때문에 전화 드렸어요.",
  "hypothesis": "예, 안녕하세요. 결제 때문에 전화드렸어요.",
  "status": "success"
 },
 {
  "file": "S00000365_0003.wav",
  "duration": 1.690,
  "latency": 1.482,
  "rtf": 0.877,
  "wer": 0.4,
  "cer": 0.4,
  "truth": "n/ 어/ 분당 살 때",
  "hypothesis": "분당 살 때",
  "status": "success"
 }
 {
  "file": "S00000365_0005.wav",
  "duration": 5.19,
  "latency": 0.866,
  "rtf": 0.167,
  "wer": 0.182,
  "cer": 0.051,
  "truth": "지금은 학습지 이용 안 하고 있거든요. 그리고 저희 서초로 이사도 했고
요.",
  "hypothesis": "지금은 학습지 이용 안 하고 있거든요. 그리고 저희 서초로 이사 또
했고요.",
  "status": "success"
 }
]
```

변환 속도 (Latency)

항목	측정값	비고
평균 변환 시간	**1.2313**	음성 파일 전송 및 API 응답 포함
p95 변환 시간	**2.3645**	네트워크 지연 상황 고려
RTF (Real-Time Factor)	**0.3964**	10초 음성을 약 4초 만에 처리

텍스트 정확도 (Accuracy)

지표	결과	목표치	달성 여부
WER (단어 오류율)	**0.2576**	≤ 0.3	**O**
CER (음절 오류율)	**0.1167**	≤ 0.15	**O**

*목표치(WER ≤ 0.3, CER ≤ 0.15)는 한국어의 교착어적 특성과 실제 통화 환경의 기계음/억양 등 데이터 난이도를 고려하여 설정함

5. 종합 평가

분류	항목	평가
RAG	실시간성	상담 흐름을 방해하지 않는 수준
RAG	검색 안정성	모든 테스트 케이스에서 정답 문서 검색
RAG	생성 신뢰성	할루시네이션 최소화
RAG	실무 활용성	상담원 보조 시스템으로 활용 가능
STT	정확성	보정 로직 없이 약 90%의 정확성을 보임 client에서 VAD 로직을 추가했기에 더 높은 성능 기대 가능
STT	효율성	빠른 변환 속도로 실시간 상담 지원에 최적화
STT	신뢰성	억지스러운 문장 생성 최소화
VAD	실시간성	
VAD	정확성	

6. 한계점 및 개선 필요 사항

💡 본 테스트는 실시간 상담 환경을 가정한 기능 및 성능 검증에 초점을 두었으나, 다음과 같은 한계가 존재한다.

- RAG 성능 평가는 12개의 대표 질의 기반으로 수행되어, 대규모 질의 분포에 대한 일반화에는 한계가 있다.
- STT 정확도 평가는 제한된 수의 음성 샘플을 기준으로 진행되어, 다양한 발화 억양·잡음 환경을 충분히 반영하지 못하였다.
- Redis 캐시는 단일 노드 기준으로 검증되어, 멀티 인스턴스 환경에서의 캐시 일관성 문제는 추가 검증이 필요하다.

---

7. 향후 개선 및 확장 계획

향후 본 시스템은 다음과 같은 방향으로 개선 및 확장을 계획하고 있다.

- 실시간 트래픽 환경을 가정한 부하 테스트 및 캐시 적중률 기반 비용/지연 최적화
- 상담 종료 후 로그를 활용한 상담 품질 피드백 및 자동 요약 고도화 기능 확장