

모델링 및 평가 테스트 계획 및 결과 보고서 [1팀]

산출물 단계	모델링 및 평가
평가 산출물	테스트 계획 및 결과 보고서
제출 일자	2025.10.01
깃허브 경로	SKN14-Final-1Team
작성 팀원	정민영, 안윤지

<개요>

본 문서는 Django 기반 LLM 활용 어플리케이션의 전체 동작을 사용자의 어플리케이션 활용 흐름에 따라 테스트하기 위한 계획서 및 결과 보고서이다.

LLM은 크게 총 두 가지로, API 문서 챗봇 모델과 사내내부문서 챗봇 모델이다.

각 모델별로 다음의 과정에 따라 어플리케이션의 동작을 검증한다.

1. API 문서 챗봇 모델 (LangGraph 기반)

1.1 모델 개요

- 기능: API 관련 질문에 대한 전문적인 답변 제공
- 구조: LangGraph 워크플로우 기반 RAG 시스템
- 주요 노드: 이미지 분석, 질문 분류, 쿼리 추출, 벡터DB 검색, 답변 생성,

답변 품질 평가, 대체 질문 생성

1.2 테스트 시나리오

<시나리오 1>: 기본 텍스트 질문 처리 (API 관련 질문)

목적: API 관련 질문에 대한 정확한 답변 생성 검증

입력 데이터:

- 사용자 질문: "bigquery의 insert 방법 알려줘"
- 세션 ID: 유효한 채팅 세션
- 대화 히스토리: 이전 대화 내용 (최대 4개)

<예상 처리 과정>:

1. analyze_image 노드: 이미지 없음으로 건너뛰기
2. classify 노드: "api" 분류 결과
3. extract_queries 노드: 이전 대화내역 + 이번 질문 통합
4. split_queries 노드: 통합된 질문으로 검색용 쿼리 분리
5. tool 노드: 분리된 쿼리 각각 벡터 DB 검색
6. basic 노드: 검색 결과를 바탕으로 답변 생성
7. evaluate 노드: 답변 평가 후 'good'이면 종료

<시나리오 2>: 이미지 포함 질문 처리

목적: 이미지와 텍스트를 결합한 질문 처리 검증

입력 데이터:

- 사용자 질문: "해결 방법 알려줘"
- 이미지 파일: MAP API 오류 스크린샷
- 세션 ID: 유효한 채팅 세션

<예상 처리 과정>:

1. **analyze_image** 노드: GPT-4o로 이미지 분석
2. **classify** 노드: 이미지 분석 결과 포함하여 분류
3. **extract_queries** 노드: 이전 대화내역 + 이번 질문 + 이미지 분석 결과 포함하여 질문 통합 및 쿼리 추출
4. **split_queries** 노드: 검색용 쿼리 분리
5. **tool** 노드: 분리된 쿼리 각각 벡터 DB 검색
6. **basic** 노드: 검색 결과를 바탕으로 이미지 분석 결과 참고하여 답변
7. **evaluate** 노드: 답변 평가 후 'good'이면 종료

<시나리오 3>: 일상 질문 처리

목적: API 외 질문에 대한 적절한 응답 검증

입력 데이터:

- 사용자 질문: "오늘 기분이 안 좋은데 뭘 하면 좋을까?"

<예상 처리 과정>:

1. **classify** 노드: "basic" 분류 결과
2. **simple** 노드: 일상 질문 답변 생성

<시나리오 4>: 전문 지식 질문 처리

목적: API 외 전문 지식 질문에 대한 거부 응답 검증

입력 데이터:

- 사용자 질문: "딥러닝에서 Transformer 구조가 뭐야?"

<예상 처리 과정>:

1. classify 노드: "none" 분류 결과
2. impossible 노드: 거부 메시지 생성

<시나리오 5>: RAG 실패 후 재검색 처리

목적: 답변 품질 평가 실패 시 대체 쿼리 생성 및 재검색 검증

입력 데이터:

- 사용자 질문: "빅쿼리에서 시트의 열을 가져오는 방법"

<예상 처리 과정>:

1. analyze_image 노드: 이미지 없음으로 건너뛰기
2. classify 노드: "api" 분류 결과
3. extract_queries 노드: 이전 대화내역 + 이번 질문 통합
4. split_queries 노드: 통합된 질문으로 검색용 쿼리 분리
5. tool 노드: 분리된 쿼리 각각 벡터 DB 검색
6. basic 노드: 검색 결과를 바탕으로 답변 생성
7. evaluate 노드: 답변 품질 평가 결과 bad
8. generate_queries 노드: 대체 답변(한글/영문) 생성
9. tool 노드: 새롭게 생성된 대체 답변으로 벡터 DB 검색
10. basic 노드: 두 검색 결과를 바탕으로 답변 생성
11. evaluate 노드: 재실행 결과는 good/bad여도 'final'로 종료 (재실행 1번 제한)

<시나리오 6>: 다중 API 질문 분리 처리

목적: 한 질문에 여러 API 요청이 포함된 경우, 쿼리 분리 및 각각 처리 검증

입력 데이터:

- 사용자 질문: "Gmail API로 메일 보내고, Google Calendar에 일정도 추가할 수 있어?"

<예상 처리 과정>:

1. **analyze_image** 노드: 이미지 없음으로 건너뛰기
2. **classify** 노드: "api" 분류 결과
3. **extract_queries** 노드: 이전 대화내역 + 이번 질문 통합
4. **split_queries** 노드: 통합된 질문으로 검색용 쿼리 분리
5. **tool** 노드 (**tool_based_search_node**): LLM이 각 쿼리마다 반드시 **vector_search_tool** 호출
 - 호출 시, 질문의 내용과 가장 관련성이 높은 API 태그를 선택
 - Gmail 관련 질문: **api_tags=["gmail"]**
 - Calendar 관련 질문: **api_tags=["calendar"]**
 - 선택된 태그를 기반으로 하이브리드 벡터 DB 검색 수행
6. **basic** 노드: Gmail API 검색 결과와 Calendar API 검색 결과를 통합하여 최종 답변 생성
7. **evaluate** 노드: 답변 평가 후 'good'이면 종료

1.3 테스트 결과

<테스트 케이스 1>: 기본 텍스트 질문

입력	"bigquery의 insert 방법 알려줘"
예상 출력	벡터 DB 검색 후 전문적인 RAG 답변
실제 출력	정확한 답변 생성
결과	성공

bigquery의 insert 방법 알려줘

BigQuery에서 데이터를 삽입하는 방법에는 여러 가지가 있습니다. 가장 일반적인 방법 중 하나는 insertAll 메서드를 사용하는 것입니다. 이 메서드는 데이터를 한 번에 여러 레코드로 스트리밍하여 삽입할 수 있게 해줍니다. insertAll 메서드를 사용하면 로드 작업을 실행할 필요 없이 실시간 데이터 처리가 가능합니다.

또한, 데이터를 스트리밍하기 위해서는 다음과 같은 HTTP POST 요청을 사용할 수 있습니다:

```
POST
https://bigquery.googleapis.com/bigquery/v2/projects/{projectId}/datasets/{datasetId}/tables/{tableId}/insertAll
```

여기서 {projectId}, {datasetId}, {tableId}는 각각 목적지의 프로젝트 ID, 데이터셋 ID, 테이블 ID를 나타냅니다.

이 외에도 다양한 프로그래밍 언어를 사용하여 BigQuery에 데이터를 삽입할 수 있으며, 각 언어에 맞는 클라이언트 라이브러리를 사용하여 데이터를 삽입할 수 있습니다. 예를 들어, Java, Node.js, Python 등의 언어로 BigQuery 클라이언트 라이브러리를 사용하여 데이터를 삽입할 수 있습니다. 각 언어에 대한 자세한 설정 방법은 해당 언어의 BigQuery API 참고 문서를 확인하시면 됩니다.

```
run_langgraph 호출 - 입력: bigquery의 insert 방법 알려줘, 이미지: False
analyze_image 호출됨 - 이미지 존재: False
[tool_based_search_node] 실행 - queries=['BigQuery에서 데이터를 삽입하는 방법은 무엇인가요?', 'How can I insert data into BigQuery?']
[vector_search_tool] hybrid 검색 완료: 'BigQuery에서 데이터를 삽입하는 방법은 무엇인가요?', tags=['bigquery']
[vector_search_tool] hybrid 검색 완료: 'How can I insert data into BigQuery?', tags=['bigquery']
[basic_langgraph_node] 생성된 답변: BigQuery에서 데이터를 삽입하는 방법에는 여러 가지가 있습니다. 가장 일반적인 방법 중 하나는 `insertAll` 메서드를 사용하는 것입니다. 이 메서드는 데이터를 한 번에 여러 레코드로 스트리밍하여 삽입할 수 있게 해줍니다. `insertAll` 메서드를 사용하면 로드 작업을 실행할 필요 없이 실시간 데이터 처리가 가능합니다.

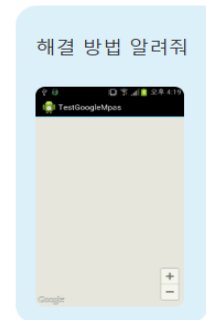
또한, 데이터를 스트리밍하기 위해서는 다음과 같은 HTTP POST 요청을 사용할 수 있습니다:

...
POST https://bigquery.googleapis.com/bigquery/v2/projects/{projectId}/datasets/{datasetId}/tables/{tableId}/insertAll
이러한 ID, 테이블 ID를 나타냅니다.
이러한 ID, 테이블 ID를 나타냅니다.

이 외에도 다양한 프로그래밍 언어를 사용하여 BigQuery에 데이터를 삽입할 수 있으며, 각 언어에 맞는 클라이언트 라이브러리를 사용하여 데이터를 삽입할 수 있습니다. 예를 들어, Java, Node.js, Python 등의 언어로 BigQuery 클라이언트 라이브러리를 사용하여 데이터를 삽입할 수 있습니다. 각 언어에 대한 자세한 설정 방법은 해당 언어의 BigQuery API 참고 문서를 확인하시면 됩니다.
[evaluate_answer_node] 최종 : good
```

<테스트 케이스 2>: 이미지 포함 질문

입력	"해결 방법 알려줘" + 이미지
예상 출력	이미지 분석 포함 + RAG 검색하여 답변
실제 출력	이미지 분석 정확 + 해결방안 제시
결과	성공



Android 디바이스에서 Google Maps가 제대로 표시되지 않고 회색 공간만 보이는 문제는 주로 인증 문제로 인해 발생할 수 있습니다. 이 문제를 해결하기 위해 다음 단계를 따라보세요:

1. **API 키 확인:** AndroidManifest.xml 파일에 올바른 API 키가 추가되었는지 확인하세요. API 키가 잘못되었거나 누락되었을 수 있습니다.
2. **결제 설정 확인:** 프로젝트에서 결제가 활성화되어 있는지 확인하세요. 결제가 활성화되지 않으면 지도 서비스가 작동하지 않을 수 있습니다.
3. **SHA-1 지문 확인:** API 키 제한에 올바른 SHA-1 디지털 지문이 추가되었는지 확인하세요. 잘못된 지문이 추가되면 인증 문제가 발생할 수 있습니다.
4. **Google Play 서비스 종속 항목 확인:** build.gradle 파일에 Google Play 서비스 종속 항목이 포함되어 있는지 확인하세요. 종속 항목이 누락되면 지도 기능이 제대로 작동하지 않을 수 있습니다.
5. **로그 확인:** Android 스튜디오의 터미널에서 `adb logcat -e "Google Maps Android API"` 명령어를 실행하여 로그를 확인하세요. 로그에서 API 키 관련 오류 메시지를 찾을 수 있습니다.

이 단계를 통해 문제를 해결할 수 있습니다. 문제가 지속되면 추가적인 로그 분석이 필요할 수 있습니다.

```
run_langraph 호출 - 입력: 해결 방법 알려줘, 이미지: True
analyze_image 호출됨 - 이미지 존재: True
[tool_based_search_node] 실행 - queries=['Android 디바이스에서 Google Maps 애플리케이션이 실행되지 않는 문제의 해결 방법은 무엇인가요?', 'what is
```

1. ****API 키 확인****: AndroidManifest.xml 파일에 올바른 API 키가 추가되었는지 확인하세요. API 키가 잘못되었거나 누락되었을 수 있습니다.
2. ****결제 설정 확인****: 프로젝트에서 결제가 활성화되어 있는지 확인하세요. 결제가 활성화되지 않으면 지도 서비스가 작동하지 않을 수 있습니다.
3. ****SHA-1 지문 확인****: API 키 제한에 올바른 SHA-1 디지털 지문이 추가되었는지 확인하세요. 잘못된 지문이 추가되면 인증 문제가 발생할 수 있습니다.
4. ****Google Play 서비스 종속 항목 확인****: build.gradle 파일에 Google Play 서비스 종속 항목이 포함되어 있는지 확인하세요. 종속 항목이 누락되면 지도 기능이 제대로 작동하지 않을 수 있습니다.
5. ****로그 확인****: Android 스튜디오의 터미널에서 `adb logcat -e "Google Maps Android API"` 명령어를 실행하여 로그를 확인하세요. 로그에서 API 키 관련 오류 메시지를 찾을 수 있습니다.

이 단계를 통해 문제를 해결할 수 있습니다. 문제가 지속되면 추가적인 로그 분석이 필요할 수 있습니다.
[evaluate_answer_node] 최종 : good

<테스트 케이스 3>: 일상 질문

입력	"오늘 기분이 안 좋은데 뭘 하면 좋을까?"
예상 출력	적절한 일상 응답
실제 출력	분류 정확 간단하고 자연스러운 일상 응답 생성
결과	성공

오늘 기분이 안 좋은데 뭘 하면 좋을까?

기분이 안 좋을 때는 산책을 하거나 좋아하는 음악을 듣는 것이 좋을 수 있어요. 친구와 이야기하거나 책을 읽는 것도 도움이 될 수 있습니다. 자신에게 맞는 방법을 찾아보세요!

기분 전환을 위한 다른 방법은?

산책의 효과는 어떤가요?

좋아하는 음악 추천은?

친구와의 대화 주제는?

책 읽기의 장점은 무엇인가요?

```
run_langraph 호출 - 입력: 오늘 기분이 안 좋은데 뭘 하면 좋을까?, 이미지: False
analyze_image 호출됨 - 이미지 존재: False
일상 질문 답변 노드 시작
```


<테스트 케이스 4>: 구글 API 이외의 전문 지식 질문

입력	"딥러닝에서 Transformer 구조가 뭐야?"
예상 출력	거부 메시지
실제 출력	분류 정확 거부 응답 답변 생성
결과	성공

딥러닝에서 Transformer 구조가 뭐야?

"딥러닝에서 Transformer 구조가 뭐야?"는 제가 모르는 내용입니다. 일상 질문 혹은 구글 API 관련 질문만 답변드릴 수 있어요.

Transformer의 기본 개념은 무엇인가?

Transformer의 주요 구성 요소는?

Transformer의 학습 과정은 어떻게 되나?

Transformer의 활용 사례는 어떤 것들이 있나?

Transformer 구조의 한계는 무엇인가?

```
run_langgraph 호출 - 입력: 딥러닝에서 Transformer 구조가 뭐야?, 이미지: False
analyze_image 호출됨 - 이미지 존재: False
답변 불가 노드 시작
```

<테스트 케이스 5>: RAG 실패 후 재검색 처리

입력	"빅쿼리에서 시트의 열을 가져오는 방법"
예상 출력	첫 답변은 부정확(bad) -> 대체 쿼리 생성 -> 재검색 후 개선된 답변
실제 출력	1차 답변: 품질 낮음 -> evaluate에서 bad 판정 2차 답변: 대체 쿼리 기반으로 검색 후 답변
결과	성공

빅쿼리에서 시트의 열을 가져오는 방법

BigQuery에서 Google Sheets의 특정 시트의 열을 가져오려면, Google Sheets API를 사용하여 해당 시트의 데이터를 가져와야 합니다. 이를 위해 `spreadsheets.values.get` 메서드를 사용할 수 있습니다. 이 메서드는 스프레드시트 ID와 범위를 지정하여 해당 범위의 데이터를 반환합니다. 예를 들어, 특정 시트의 A1부터 B20까지의 데이터를 가져오려면 다음과 같은 HTTP GET 요청을 사용할 수 있습니다:

```
GET https://sheets.googleapis.com/v4/spreadsheets/{spreadsheetId}/values/{range}
```

여기서 `{spreadsheetId}`는 데이터를 가져올 스프레드시트의 ID이고, `{range}`는 값을 검색할 범위의 A1 표기법입니다. 예를 들어, `Sheet1!A1:B20`과 같은 형식으로 범위를 지정할 수 있습니다. 이 요청을 통해 시트의 열 데이터를 가져올 수 있습니다.

```
run_langgraph 호출 - 입력: 빅쿼리에서 시트의 열을 가져오는 방법, 이미지: False
analyze_image 호출됨 - 이미지 존재: False
[tool_based_search_node] 실행 - queries=['빅쿼리에서 시트의 열을 가져오는 방법은 무엇인가요?', 'How can I retrieve columns from a sheet in BigQuery?']
[vector_search_tool] hybrid 검색 완료: '빅쿼리에서 시트의 열을 가져오는 방법은 무엇인가요?', tags=['bigquery', 'sheets']
[vector_search_tool] hybrid 검색 완료: 'How can I retrieve columns from a sheet in BigQuery?', tags=['bigquery', 'sheets']
[basic_langgraph_node] 생성된 답변: BigQuery에서 시트의 열을 가져오는 방법에 대한 정보는 제공된 문서에 명시되어 있지 않습니다. 따라서 이 질문에 대한 답변을 제공
와 관련된 다른 질문이 있으면 말씀해 주세요.
[evaluate_answer_node] 최종 bad
[generate_alternative_queries] 생성된 쿼리: ['Google Sheets API를 사용하여 Google Sheets의 데이터를 BigQuery로 가져올 수 있습니다. 먼저, Google Sheets API를 통해
합니다. 그런 다음, BigQuery API를 사용하여 해당 데이터를 BigQuery 테이블에 로드할 수 있습니다. 이를 위해 Google Cloud Console에서 BigQuery와 Sheets API를 활성화
설정해야 합니다.', 'You can use the Google Sheets API to import data from Google Sheets into BigQuery. First, use the Google Sheets API to read the data from th
igQuery API to load this data into a BigQuery table. To do this, you need to enable both the BigQuery and Sheets APIs in the Google Cloud Console and set up the
[tool_based_search_node] 실행 - queries=['Google Sheets API를 사용하여 Google Sheets의 데이터를 BigQuery로 가져올 수 있습니다. 먼저, Google Sheets API를 통해 시
입니다. 그런 다음, BigQuery API를 사용하여 해당 데이터를 BigQuery 테이블에 로드할 수 있습니다. 이를 위해 Google Cloud Console에서 BigQuery와 Sheets API를 활성화하
정해야 합니다.', 'You can use the Google Sheets API to import data from Google Sheets into BigQuery. First, use the Google Sheets API to read the data from the s
query API to load this data into a BigQuery table. To do this, you need to enable both the BigQuery and Sheets APIs in the Google Cloud Console and set up the nee
[vector_search_tool] hybrid 검색 완료: 'Google Sheets API를 사용하여 Google sheets의 데이터를 BigQuery로 가져 올 수 있습니다. 먼저, Google Sheets API를 통해 시트
다. 그런 다음, BigQuery API를 사용하여 해당 데이터를 BigQuery 테이블에 로드할 수 있습니다. 이를 위해 Google Cloud Console에서 BigQuery와 Sheets API를 활성화하고
해야 합니다.', tags=['sheets', 'bigquery']
[vector_search_tool] hybrid 검색 완료: 'You can use the Google Sheets API to import data from Google Sheets into BigQuery. First, use the Google Sheets API to re
et. Then, use the BigQuery API to load this data into a BigQuery table. To do this, you need to enable both the BigQuery and Sheets APIs in the Google Cloud Con
sary credentials.', tags=['sheets', 'bigquery']
[basic_langgraph_node] 생성된 답변: BigQuery에서 Google Sheets의 특정 시트의 열을 가져오려면, Google Sheets API를 사용하여 해당 시트의 데이터를 가져와야 합니다.
values.get' 메서드를 사용할 수 있습니다. 이 메서드는 스프레드시트 ID와 범위를 지정하여 해당 범위의 데이터를 반환합니다. 예를 들어, 특정 시트의 A1부터 B20까지의
과 같은 HTTP GET 요청을 사용할 수 있습니다:
...
GET https://sheets.googleapis.com/v4/spreadsheets/{spreadsheetId}/values/{range}
...
여기서 '{spreadsheetId}'는 데이터를 가져올 스프레드시트의 ID이고, '{range}'는 값을 검색할 범위의 A1 표기법입니다. 예를 들어, 'Sheet1!A1:B20'과 같은 형식으로 범위
이 요청을 통해 시트의 열 데이터를 가져올 수 있습니다.
[evaluate_answer_node] 최종 good
```

<테스트 케이스 6>: 다중 쿼리 질문

입력	"Gmail API로 메일 보내고, Google Calendar에 일정도 추가할 수 있어?"
예상 출력	Gmail/Calendar 각각 쿼리 분리 api_tags=["gmail"], api_tags=["calendar"] 선택 후 검색 두 결과를 통합하여 최종 답변
실제 출력	Gmail API 메일 전송 방법 + Calendar API 일정 추가 방법 모두 포함된 답변
결과	성공
비고	다중 쿼리 분리 및 태그 선택 정상

Gmail API로 메일 보내고, Google Calendar에 일정도 추가할 수 있어?

네, Gmail API를 사용하여 이메일을 보낼 수 있으며, Google Calendar API를 사용하여 일정을 추가할 수 있습니다.

1. **Gmail API로 메일 보내기:** Gmail API를 사용하여 이메일 보내는 방법은 두 가지가 있습니다. `messages.send` 메서드를 사용하여 직접 전송하거나, `drafts.send` 메서드를 사용하여 초안에서 보낼 수 있습니다. 이메일은 base64url로 인코딩된 문자열로 전송됩니다.
2. **Google Calendar에 일정 추가하기:** Google Calendar API를 사용하여 일정을 추가하려면 `POST` 메서드를 사용하여 `/calendars/calendarId/events` 엔드포인트에 요청을 보내야 합니다. 요청 본문에는 일정의 세부 정보가 포함되어야 합니다.

이 두 가지 기능을 통해 Gmail과 Google Calendar를 효과적으로 활용할 수 있습니다.

```
run_langgraph 호출 - 입력: Gmail API로 메일 보내고, Google Calendar에 일정도 추가할 수 있어?, 이미지: False
analyze_image 호출됨 - 이미지 존재: False
[tool_based_search_node] 실행 - queries=['Gmail API로 메일을 보내고 Google Calendar에 일정을 추가할 수 있나요?', 'Is it possible to send emails using the Gmail API and add events to Google Calendar?']
[vector_search_tool] hybrid 검색 완료: 'Gmail API로 메일을 보내고 Google Calendar에 일정을 추가할 수 있나요?', tags=['gmail', 'calendar']
[vector_search_tool] hybrid 검색 완료: 'Is it possible to send emails using the Gmail API and add events to Google Calendar?', tags=['gmail', 'calendar']
[basic_langgraph_node] 생성된 답변: 네, Gmail API를 사용하여 이메일을 보낼 수 있으며, Google Calendar API를 사용하여 일정을 추가할 수 있습니다.

1. **Gmail API로 메일 보내기**: Gmail API를 사용하여 이메일을 보내는 방법은 두 가지가 있습니다. `messages.send` 메서드를 사용하여 직접 전송하거나, `drafts.send` 메서드를 사용하여 초안에서 보낼 수 있습니다. 이메일은 base64url로 인코딩된 문자열로 전송됩니다.

2. **Google Calendar에 일정 추가하기**: Google Calendar API를 사용하여 일정을 추가하려면 `POST` 메서드를 사용하여 `/calendars/calendarId/events` 엔드포인트에 요청을 보내야 합니다. 요청 본문에는 일정의 세부 정보가 포함되어야 합니다.

이 두 가지 기능을 통해 Gmail과 Google Calendar를 효과적으로 활용할 수 있습니다.
[evaluate_answer_node] 최종 : good
```

1-2. 음성 인식 모델 (Whisper 기반)

1-2.1 모델 개요

- 기능: 음성 파일을 텍스트로 변환 후 챗봇 처리
- 구조: Open AI Whisper API => 텍스트로 변환 후 챗봇 LangGraph로 전달
- 특징: 다양한 품질의 음성 인식 최적화

1-2.2 테스트 시나리오

<시나리오>: 사용자가 음성으로 입력하는 경우

목적: 음성 파일의 정확한 텍스트 변환 검증

입력 데이터:

- 음성 파일: "bigquery의 인서트 방법 알려줘" (한국어)
- 세션 ID: 유효한 채팅 세션

예상 처리 과정:

1. call_whisper_api(audio_file) 호출
2. Whisper API로 음성을 텍스트 변환
3. 변환된 텍스트로 LangGraph 실행
4. 답변 생성 및 저장

1-2.3 테스트 결과

<테스트 케이스>: 정상 음성 인식

입력	"bigquery의 insert 방법 알려줘"
예상 출력	정확한 텍스트 변환
실제 출력	정확한 텍스트 변환
결과	성공
비고	인식을 우수

빅쿼리의 인서트 방법 알려줘

BigQuery에서 데이터를 삽입하는 방법에는 여러 가지가 있습니다. 비동기 작업을 시작하려면 'jobs.insert' 메서드를 사용하고, 데이터를 한 번에 여러 레코드로 삽입하려면 'insertAll' 메서드를 사용할 수 있습니다. 'insertAll' 메서드는 데이터를 로드 작업 없이 스트리밍 방식으로 BigQuery에 삽입할 수 있어 실시간 데이터 처리가 가능합니다. 각 메서드에 대한 자세한 정보는 다음 링크에서 확인할 수 있습니다: - [jobs.insert 메서드] (<https://cloud.google.com/bigquery/docs/reference/rest/v2/jobs/insert>) - [insertAll 메서드] (<https://cloud.google.com/bigquery/docs/reference/rest/v2/tabledata/insertAll>)

[illegible]

1-3. 통합 결과

<전체 워크플로우 테스트 시나리오 및 결과>

목적: 사용자 시나리오에 따른 전체 시스템 동작 검증

<시나리오>: API 문서 챗봇 사용자 여정

1. 로그인 → 사용자 인증
2. 새 채팅 세션 생성 → 세션 ID 발급
3. 텍스트 질문 → "빅쿼리 insert 방법 알려줘"
4. 이미지 업로드 → 구글 drive 관련 API 코드 스크린샷
5. 이미지와 함께 텍스트 질문 → "이 이미지는 무슨 코드야?"
6. 대화 히스토리 확인 → "내가 방금 뭐 물어봤어?" → 연속성 검증
7. 음성 질문 → "BigQuery에서 데이터셋을 삭제하려면?" → 음성 인식 후 답변

<테스트 결과: 모든 단계에서 정상 응답>

빅쿼리의 insert 방법 알려줘

BigQuery에서 데이터를 삽입하는 방법은 여러 가지가 있습니다. 비동기 작업을 시작하려면 'jobs.insert' 메서드를 사용하고, 데이터를 한 번에 여러 레코드로 삽입하려면 'insertAll' 메서드를 사용할 수 있습니다. - 비동기 작업을 시작하려면 'insert' 메서드를 사용합니다. 자세한 내용은 [여기](https://cloud.google.com/bigquery/docs/reference/rest/v2/jobs/insert)를 참조하세요. - 데이터를 여러 레코드로 삽입하려면 'insertAll' 메서드를 사용하여 데이터를 스트리밍할 수 있습니다. 자세한 내용은 [여기](https://cloud.google.com/bigquery/docs/reference/rest/v2/tabledata/insertAll)를 참조하세요. 이 외의 방법이나 추가적인 정보는 제공된 문서에 포함되어 있지 않습니다.

이 이미지는 무슨 코드야?



이미지에 대한 질문을 하셨습니다. 제공된 이미지는 Google Drive API의 'about.get' 메소드와 관련된 정보가 나와 있습니다. 이 메소드는 Drive API 설정과 함께 현재 사용자에게 대한 정보를 가져오는 데 사용됩니다. 이미지의 주요 내용은 다음과 같습니다: - **HTTP 요청**: 'GET https://www.googleapis.com/drive/v2/about' URL이 표시되어 있습니다. - **쿼리 매개변수**: - 'includeSubscribed': Boolean 타입으로, 특정 계정 외부의 변경 사항을 계산할지 여부를 결정합니다. - 'maxChangelistCount': String 타입으로, 계산할 최대 변경 ID의 개수를 지정합니다. - 'startChangelist': String 타입으로, 남은 변경 ID 수를 계산할 때부터 시작하도록 변경 ID를 지정합니다. 이 정보는 Google Drive API를 사용하여 사용자 정보를 가져오는 방법에 대한 것입니다. 추가적인 설명으로는 요청 본문이 비어 있어야 하며, 응답 본문에 대한 정보도 제공된다는 점이 있습니다.

내가 방금 뭐 물어봤어?

사용자님께서 방금 물어보신 질문은 "내가 방금 뭐 물어봤어?"입니다. 이전에는 Google Drive API의 'about.get' 메소드와 관련된 이미지에 대해 설명을 요청하셨습니다.

Google Drive API란 무엇인가요?

'about.get' 메소드의 사용 예시

Google Drive API의 오류 해결 방법

Google Drive API의 보편 시례는?

API 호출 시 주의사항은 무엇인가요?

BigQuery에서 데이터셋을 삭제하려면?

BigQuery에서 데이터셋을 삭제하려면 'delete' 메서드를 사용해야 합니다. 요청은 다음과 같이 구성됩니다: "" DELETE /bigquery/v2/projects/{projectId}/datasets/{datasetId} "" 여기서 '{projectId}'는 데이터셋이 포함된 프로젝트의 ID이고, '{datasetId}'는 삭제할 데이터셋의 ID입니다. 데이터셋을 삭제하기 전에 해당 데이터셋의 모든 테이블을 삭제해야 하며, 이는 수동으로 삭제하거나 'deleteContents' 파라미터를 사용하여 삭제할 수 있습니다. 데이터셋 삭제를 위해서는 다음 중 하나의 OAuth 스크립트가 필요합니다: - 'https://www.googleapis.com/auth/bigquery' - 'https://www.googleapis.com/auth/cloud-platform'

데이터셋 삭제 전 확인 사항은?

OAuth 스크립트 설정 방법은?

데이터셋 삭제 단계는 어떻게 되나?

테이블 삭제 자동화 방법은?

보통 사례로 데이터셋 관리 방법은?

2. [API 문서 챗봇] 랭그래프 답변 평가

2.1 QA vs 원문 vs QA + 원문 하이브리드

비교 조건

- 동일 합성 데이터셋
 - RAGAS를 통해 생성한 20개 합성 질문셋 ([dataset2.csv](#))
 - 페르소나 정의: Expert, Beginner
- 동일 평가 모델 : gpt-4.1-mini
- 동일 평가 지표
 - **Context Recall**: 검색된 문맥이 질문과 기준 정답을 얼마나 잘 포괄하는지
 - **Faithfulness**: 생성된 답변이 검색된 문맥에 얼마나 충실한지
 - **Factual Correctness (mode = f1)**: 생성된 답변이 기준 정답과 의미적으로 얼마나 일치하는지 (F1 스코어 기반)

평가 대상

- 동일 랭그래프 특징: 벡터 db 검색을 tool call 통해서 진행하는 구조, 분류 기준 구체화

평가 결과

버전	Context Recall	Faithfulness	Factual Correctness
원문	0.6500	0.6013	0.3475
QA	0.7500	0.7205	0.3130
원문+ QA	0.8000	0.6912	0.3185

분석

- 원문 기반 RAG
 - Recall: 0.6500, Faithfulness: 0.6013, Correctness: 0.3475
 - 문맥 회수율(Recall)이 낮고 답변 충실성도 제한적이나, Correctness는 상대적으로 안정적
- QA 기반 RAG
 - Recall: 0.7500, Faithfulness: 0.7205, Correctness: 0.3130
 - Recall과 Faithfulness는 개선되었으나 Correctness가 가장 낮음
 - 즉, 검색은 더 잘하지만 정답과의 의미적 일치는 다소 부족
- 원문 + QA 하이브리드
 - Recall: 0.8000, Faithfulness: 0.6912, Correctness: 0.3185
 - Recall은 세 구조 중 가장 높음 → 질문을 더 폭넓게 커버
 - Faithfulness와 Correctness는 중간 수준으로 균형을 보임

결론

평가 결과, QA 기반 RAG는 질문의 맥락을 잘 회수하나 정답 일치율이 낮고, 원문 기반 RAG는 Correctness는 상대적으로 높지만 Recall이 낮아 보완이 필요했다. 두 구조가 잘 처리하는 질문 유형이 상호 보완적이었다.

QA 벡터 DB와 원문 벡터 DB를 모두 검색하는 하이브리드 방식은 Recall 측면에서 가장 우수하며, 답변 충실성과 정확성에서도 균형 잡힌 결과를 보여준다. 따라서 QA 벡터 DB와 원문 벡터 DB를 모두 검색하는 하이브리드 방식을 선택하였다.

2.2 QA + 원문 벡터 DB 하이브리드

비교 조건

- 동일 합성 데이터셋
 - RAGAS를 통해 생성한 20개 합성 질문셋 ([dataset2.csv](#))
- 동일 평가 모델 : gpt-4.1-mini
- 동일 평가 지표
 - [RAGAS]
 - Context Recall: 검색된 문맥이 질문과 기준 정답을 얼마나 잘 포괄하는지
 - Faithfulness: 생성된 답변이 검색된 문맥에 얼마나 충실한지
 - Factual Correctness (mode = f1): 생성된 답변이 기준 정답과 의미적으로 얼마나 일치하는지 (F1 스코어 기반)
 - [자체 정량 평가]
 - 정답 정확성 (Answer Correctness): API 계약·권장사항 기준으로 답변의 정확성을 평가 (1~5점 척도)
 - 근거 충실성 (Groundedness): 생성된 답변이 검색된 컨텍스트에 얼마나 충실히 기반하는지 평가 (1~5점 척도)
 - 컨텍스트 충분성 (Context Coverage): 검색된 컨텍스트가 답변을 뒷받침하기에 충분한지 평가 (1~5점 척도)

평가 대상

- VER1 (utils_sec) : 원문 벡터 DB + QA 벡터 DB 하이브리드 (하나의 프롬프트에 같이 주는 방식)
- VER2 (utils_third) : 원문 벡터 DB → QA 벡터 DB 순차 하이브리드
- VER3 (utils_v4) : 원문 벡터 DB + QA 벡터 DB → 답변 평가 후 → Top-K 늘리기
- VER4 (utils_v5) : 원문 벡터 DB + QA 벡터 DB → 답변 평가 후 → HyDE 재실행
- VER5 (utils_v6) : 원문 벡터 DB + QA 벡터 DB → 답변 평가 후 → 2개의 질문 재생성 → 재실행
- VER6 (utils_v7) : 원문 벡터 DB + QA 벡터 DB → 답변 평가 후 → 3개의 검색 쿼리 재생성 → 재실행
- VER7 (utils_v8) : 원문 벡터 DB + QA 벡터 DB (앙상블) → 답변 평가 후 → 3개의 검색 쿼리 재생성 → 재실행
- VER8 (utils_sec5) : 쿼리 정제 시 영어+한글 2가지 질문 생성, HyDE 가상 답변도 영어+한글 2가지로 생성, dense 검색만 사용, 재시도 시 원문 10개 / QA 40개
- VER9 (utils_final) : 최종 모델 - 쿼리 정제 시 영어+한글 2가지 질문 생성, HyDE 가상 답변도 영어+한글 2가지로 생성, 원문 벡터 DB + QA 벡터 DB (앙상블) +[원문/QA top-k] 5, 20 => 15, 30

[RAGAS] 평가 결과

버전	특징	Context Recall	Faithfulness	Factual Correctness
VER1	QA+원문 단순 병합	0.8000	0.6912	0.3185
VER2	원문→QA 순차 검색	0.7500	0.6658	0.2930
VER3	답변 평가 후 Top-K 증가	0.6750	0.6803	0.3025
VER4	답변 평가 후 HyDE 재실행	0.8500	0.8857	0.4135
VER5	답변 평가 후 2개 질문 재생성	0.7250	0.6594	0.4650
VER6	답변 평가 후 3개 검색 쿼리 재생성	0.8250	0.7624	0.5000
VER7	QA+원문 앙상블 + 3개 검색 쿼리 재생성	0.8500	0.8857	0.4135
VER8	한·영 쿼리 생성 + HyDE 다국어 가상 답변	0.8750	0.9128	0.5350
VER9	최종 앙상블 모델 (한·영 쿼리 + HyDE + QA+원문 DB)	0.9250	0.8521	0.5455

[자체 평가] 평가 결과

버전	특징	전체 평균	정답 정확성	근거 충실성	컨텍스트 충분성
VER1	QA+원문 단순 병합	72.33	71.00	75.00	71.00
VER2	원문→QA 순차 검색	65.00	63.00	67.00	65.00
VER3	답변 평가 후 Top-K 증가	63.67	63.00	65.00	63.00
VER4	답변 평가 후 HyDE 재실행	67.33	65.00	70.00	67.00
VER5	답변 평가 후 2개 질문 재생성	70.67	70.00	72.00	70.00
VER6	답변 평가 후 3개 검색 쿼리 재생성	81.33	82.00	81.00	81.00
VER7	QA+원문 앙상블 + 3개 검색 쿼리 재생성	84.67	84.00	86.00	84.00
VER8	한·영 쿼리 생성 + HyDE 다국어 가상 답변	79.33	79.00	80.00	79.00
VER9	최종 앙상블 모델 (한·영 쿼리 + HyDE + QA+원문 DB)	80.67	80.00	82.00	80.00

분석

- **RAGAS 평가 결과:**

- VER9이 Context Recall 0.9250, Factual Correctness 0.5455로 모든 버전 중 가장 높은 성능을 기록
- VER6과 VER7은 Correctness는 일정 수준 확보했으나, Recall과 Faithfulness에서 VER9보다 낮음

- **자체 정량 평가 결과:**

- VER6(답변 평가 후 3개 쿼리 재생성)과 VER7(QA+원문 양상블+3개 쿼리 재생성)이 평균 점수(81~84점)에서 가장 높게 나타남
- 그러나 추가 재실행 과정을 거쳤을 때의 결과이며, 첫 실행에서 실패하는 경우가 많아 안정성이 떨어짐
- VER9은 평균 80점으로 높은 점수를 기록

- **최종 선정 근거:**

- VER6,7은 재실행을 반복해야 안정적인 답변을 확보할 수 있어 응답 속도가 느려짐
- 반면 VER9은 첫 실행에서 실패율이 낮고, QA+원문 DB 양상블·한영 쿼리·HyDE 기반 다국어 생성까지 통합하여 빠른 속도와 안정적 성능을 동시에 보장
- 따라서 RAGAS와 자체평가 모두에서 높고, 균형 잡힌 성능을 보인 VER9을 최종 모델로 선정

2.3 top-k 테스트

비교 조건

- 동일 합성 데이터셋
 - RAGAS를 통해 생성한 20개 합성 질문셋 ([dataset2.csv](#))
- 동일 평가 모델 : gpt-4.1-mini
- 동일 평가 지표
 - [RAGAS]
 - Context Recall: 검색된 문맥이 질문과 기준 정답을 얼마나 잘 포괄하는지
 - Faithfulness: 생성된 답변이 검색된 문맥에 얼마나 충실한지
 - Factual Correctness (mode = f1): 생성된 답변이 기준 정답과 의미적으로 얼마나 일치하는지 (F1 스코어 기반)
 - [자체 정량 평가]
 - 응답 정확성 (Answer Correctness): 생성된 답변이 참조 답변과 비교하여 정확하고 완전한 정보를 제공하는지 평가 (1~5점 척도)
 - 컨텍스트 관련성 (Context Relevance): 검색된 컨텍스트가 질문에 대답하기 위해 관련성이 높은지 평가 (1~5점 척도)
 - 컨텍스트 충실성 (Context Faithfulness): 생성된 답변이 주어진 컨텍스트에만 기반하는지, 아니면 없는 정보를 추가했는지 평가 (1~5점 척도)
 - 컨텍스트 충분성 (Context Recall): 검색된 컨텍스트가 질문에 완전히 답변하기에 충분한 정보를 포함하는지 평가 (1~5점 척도)

평가 대상

- 랭그래프 구조: QA + 원문 벡터 DB 하이브리드 VER9
- 첫 실행 top-k 고정: 5 (원문), 20 (QA)

[RAGAS] 평가 결과

재 실행 top-k	Context Recall	Faithfulness	Factual Correctness
8, 15	0.9250	0.8304	0.5245
10, 30	0.9250	0.8718	0.5535
15, 30	0.9250	0.8521	0.5455
20, 30	0.9250	0.9213	0.5465

[자체 평가] 평가 결과 (20점 만점)

재 실행 top-k	평균 총점	응답 정확성 평균	컨텍스트 관련성 평균	컨텍스트 충실성 평균	컨텍스트 충분성 평균
8, 15	18.25	4.30	4.70	4.55	4.70
10, 30	18.15	4.35	4.55	4.65	4.60
15, 30	18.35	4.35	4.65	4.65	4.70
20, 30	18.15	4.35	4.55	4.65	4.60

분석

- **RAGAS 평가:**

- Top-K 조합별로 Context Recall·Faithfulness·Correctness가 모두 큰 차이를 보이지 않음
- 따라서 Top-K 최적값 결정에서는 RAGAS 점수보다는 자체 정량 평가 결과를 더 중점적으로 고려함

- **자체 정량 평가:**

- 15,30 조합이 평균 총점(18.35)에서 가장 높은 성능을 기록
- 20,30 조합은 Faithfulness(0.9213)에서 최고를 기록했지만, Correctness에서 큰 차별성이 없었고 복잡성 증가로 응답 속도 저하 가능성이 있음

- **최종 선정 근거:**

- RAGAS 점수는 조합 간 큰 차이가 없어 세부 지표 균형과 안정성을 보여준 자체 정량 평가를 기준으로 최종 선택.
- [원문/QA] 5,20 → 15,30 설정에서 균형적이고 높은 점수가 확보되어, 속도와 안정성을 모두 고려하여 Top-K로 선정

2.4 Perplexity와 비교

평가 기준

- **평가 항목:** 정확도(Accuracy), 재현율(Recall), 신뢰성(Reliability)
- **평가 방식:** 평가 기준에 따라 GPT-4.1-mini 평가 모델이 채점
- **프롬프트**

prompt = f"""

당신은 AI가 생성한 답변이 **기준 정답**과 의미적으로 얼마나 잘 일치하는지를 평가하는 전문가입니다.

이 평가는 **정답과 문장이 달라도 핵심 내용에 대한 의미가 같다면 인정**합니다.

하지만 기준 정답에 **없는 정보**, **추측**, **불필요한 코드**는 감점 요인입니다.

평가 원칙:

- 답변이 **기준 정답과 단어/문장이 다르더라도**,
핵심 내용의 의미가 같으면 높은 점수를 줘야 합니다.
- 특히 **웹 기반 배경 설명**, **예측성 문장**, **불필요한 코드 예시**는 감점합니다.
- **핵심 정보(조건, 함수명, 파라미터 등)**는 꼭 포함되어야 합니다.

평가 항목:

1. **정확도 (Accuracy):**

- 3점: 표현은 달라도 기준 정답과 **의미가 동일**
- 2점: 핵심은 일치하지만, 일부 세부 의미나 조건이 다름
- 1점: 중요한 내용은 있으나 일부 왜곡되었거나 혼동 있음
- 0점: 기준 정답과 **의미적으로도 불일치**하거나 틀림

2. **재현율 (Recall):**

- 3점: 기준 정답의 **핵심 정보(논리, 순서, 구성요소)**를 빠짐없이 포함
- 2점: 주요 정보는 있지만 일부가 빠짐
- 1점: 핵심 정보 일부만 존재하거나 흐릿하게 표현됨
- 0점: 기준 정답의 중요한 내용이 대부분 빠짐

3. **신뢰성 (Reliability)**:

- 3점: **기준 정답 안에서만 충실하게** 답변 (부가 설명 없음)
- 2점: 대체로 맞지만, 부가 설명이 약간 포함됨
- 1점: 일반 지식, 예시 코드, 추측성 문장 등 기준 외 정보 포함
- 0점: 기준 정답과 무관하거나 과한 배경설명이 많음

평가 결과

모델	평균 정확도	평균 재현율	평균 신뢰성	전체 평균
LangGraph (우리 모델)	83.33	76.67	61.67	73.89
Perplexity	66.67	58.33	33.33	52.78

분석

- LangGraph는 Perplexity 대비 정확도 +16.66, 재현율 +18.34, 신뢰성 +28.34 포인트 개선하였다.

2.5 종합 결론 및 최종 모델 선정 이유

- QA DB 단독: Recall·Faithfulness는 안정적이거나 Correctness가 낮음 → 단독 활용 부적합
- 원문 DB 단독: 문맥 회수율(Recall)은 우수하지만 Correctness 한계 존재 → 보조적 역할에 적합
- 하이브리드 구조: QA와 원문 DB 결합 시 성능이 대폭 향상
- 특히 다국어 쿼리 + HyDE 적용(VER9)에서 최고 성능 달성
- Recall 0.9250, Faithfulness 0.8521, Correctness 0.5455 → 균형 잡힌 최고 성능
- Top-K 조정: (15, 30) 설정에서 안정적 성능 확보 가능
- Perplexity와 비교: LangGraph가 모든 지표에서 우위, 특히 신뢰성에서 큰 차이

따라서, 본 평가에서는 하이브리드 최종 모델(VER9, utils_final)을 최종 선택 모델로 확정하였다.

3. [API 문서 챗봇] 결론 및 개선사항

3.1 주요 성과

1. API 문서 챗봇

- 질문 분류(api/basic/none)와 답변 생성 정확도 우수
- 다중 API 질문 분리 및 처리, 재검색(fallback)까지 정상 동작

2. 음성 인식(Whisper)

- 음성입력에 대한 우수한 인식률

3. 이미지 처리(GPT-4o)

- 사용자가 입력한 이미지에 대한 정확한 분석 결과 생성
- 연속 질문에서도 이미지 분석 결과가 반영됨

4. 오류 처리 및 예외 대응

- RAG 품질 평가(evaluate) 및 대체 쿼리 생성(generate_queries)을 통한 fallback 정상 동작

3.2 향후 계획

- 사용자 경험 개선 : 답변 스타일·길이·속도 최적화
- RAG 평가 및 성능 최적화 지속 (프롬프트, 검색 방식, LangGraph 구조)
- 배포 환경에서의 답변 속도 및 품질 최적화
- 임베딩 모델 파인튜닝 & 리랭커 모델 적용 : 답변 정확도 개선
- 벡터 DB 최신화 : 자동 크롤링 기반 최신 문서 반영으로 최신성 확보
- BM25 개선 : 한국어 형태소 분석기를 적용하여 정확도 향상

4. 사내 내부 문서 챗봇 테스트

4.1 정량 성능 평가

TOOL CALL 멀티턴 성능 평가

평가 데이터: 학습 데이터와 동일한 방식으로 내용만 다른 테스트 데이터셋 생성

평가 기준: tool_selection, params_selection, params_value_similarity

모델	tool_selection	params_selection	params_value_similarity
Qwen3-8B (기본)	79.22%	79.22%	77.21%
Qwen2.5-7B (기본)	25.97%	25.97%	67.33%
Qwen3-8B (파인튜닝)	98.05%	98.05%	88.50%
Qwen2.5-7B (파인튜닝)	99.68%	99.68%	87.55%

1. 툴 선택 정확도 (tool_selection)

- **목표:** 정답과 예측의 TOOL NAME이 일치하는지 평가.
- **계산:** 정답에 `tool_call` 있는 샘플 중, 예측이 맞은 개수 / 총 샘플 수
- **해석:** "정확한 도구를 선택했는지" 평가.

2. 파라미터 키 선택 정확도 (params_selection)

- **목표:** 정답과 예측의 파라미터 키(TOOL CALL의 파라미터 키)들이 얼마나 일치하는지 평가.
- **계산:** 정답 키와 예측 키의 매칭 수 / (정답 키 수 + 예측에만 있는 키 수)
- **해석:** "필요한 키(TOOL CALL의 키)를 정확히 선택했는지, 불필요한 키를 추가하지 않았는지" 평가.

3. 파라미터 값 유사도 (params_value_similarity)

- **목표:** 공통 키의 값이 얼마나 유사한지 평가.
- **계산:** 형태소 Jaccard(0.6) + 문자 유사도(0.4)의 평균 유사도 계산.
- **해석:** "선택된 값(TOOL CALL 키의 값)들이 얼마나 정확하게 일치하는지" 평가.

RAG 평가

평가 방식: 자체 제작 데이터셋(55개) + RAGAS 활용

평가 기준:

- **Context Recall:** 검색된 문맥이 질문과 기준 정답을 얼마나 잘 포괄하는지
- **Faithfulness:** 생성된 답변이 검색된 문맥에 얼마나 충실한지
- **Factual Correctness (mode = f1):** 생성된 답변이 기준 정답과 의미적으로 얼마나 일치하는지 (F1 스코어 기반)

모델	context_recall	faithfulness	factual_correctness(f1)
Qwen2.5-7B (파인튜닝)	0.7697	0.7992	0.3191
Qwen3-8B (파인튜닝)	0.9273	0.8614	0.3709

평가 방식: 자체 제작 데이터셋(55개) + LLM-AS-JUDGE 자체 정량 RAG 평가

평가 기준: 정확도, 재현율, 구체성

- **정확도 (Accuracy):** 시스템 답변이 기준 정답과 얼마나 일치하는지 평가. 점수 (0~3점)
 - 3점 (완벽 일치)
 - 2점 (약간 차이)
 - 1점 (핵심 맞지만 세부 누락)
 - 0점 (큰 차이)
 - 100점 환산: $(\text{정확도 점수} / 3) * 100$
- **재현율 (Recall):** 시스템 답변이 기준 정답에서 중요한 정보를 얼마나 잘 재현했는지 평가. 점수 (0~3점)
 - 3점 (모든 핵심 정보 정확히 재현)
 - 2점 (핵심 정보는 잘 재현되었으나 세부 부족)
 - 1점 (핵심 정보 누락)
 - 0점 (핵심 정보 많이 누락)
 - 100점 환산: $(\text{재현율 점수} / 3) * 100$
- **구체성 (Specificity):** 답변에서 세부 사항(예시, 매개변수 설명 등)을 얼마나 구체적으로 설명했는지 평가. 점수 (0~3점)
 - 3점 (구체적 설명 및 예시 포함)
 - 2점 (핵심 매개변수 설명, 예시 부족)
 - 1점 (기본 설명만 제공)
 - 0점 (세부 사항 누락)
 - 100점 환산: $(\text{구체성 점수} / 3) * 100$
- **최종 평균 점수** = $(\text{정확도} + \text{재현율} + \text{구체성}) / 3$

모델	정확도	재현율	구체성	평균
Qwen2.5-7B (파인튜닝)	67.27	70.91	55.76	64.65
Qwen3-8B (파인튜닝)	70.91	74.55	65.45	70.30

TOP_K 테스트 (LLM 기반 정량적 RAG 평가)

평가 기준: 정확도, 재현율, 구체성, 최종 평균 점수

항목	목표	기준	100점 환산
정확도 (Accuracy)	시스템 답변이 기준 정답과 얼마나 일치하는지	3점: 완벽 일치 2점: 약간 차이 1점: 핵심 맞지만 세부 누락 0점: 큰 차이	$(\text{점수} / 3) * 100$
재현율 (Recall)	기준 정답의 중요한 정보를 얼마나 잘 재현했는지	3점: 모든 핵심 정보 재현 2점: 핵심 정보 재현, 세부 부족 1점: 핵심 정보 누락 0점: 많이 누락	$(\text{점수} / 3) * 100$
구체성 (Specificity)	세부 사항(매개변수, 예시 등)을 얼마나 구체적으로 설명했는지	3점: 구체적 설명 및 예시 포함 2점: 매개변수 설명, 예시 부족 1점: 기본 설명만 0점: 세부 누락	$(\text{점수} / 3) * 100$
최종 평균 점수	정확도+재현율+구체 성의 평균	-	$(\text{정확도} + \text{재현율} + \text{구체성}) / 3$

top_k 값에 따른 성능 변화:

값	평균 정확도	평균 재현율	평균 구체성	전체 평균
4	70.30	74.55	63.64	69.49
5	71.52	75.76	65.45	70.91
6	70.91	73.94	64.24	69.70
7	71.52	77.58	68.48	72.53
8	63.64	69.70	63.64	65.66

결론: k=7일 때 가장 높은 평균 성능(72.53)을 기록하였으며, 최적의 k 값으로 판단됨.

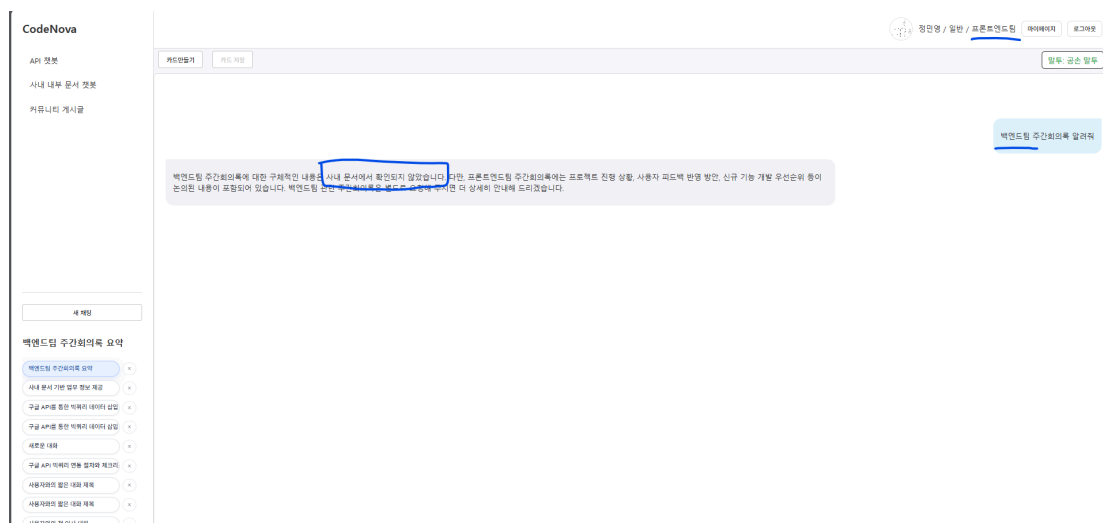
4.2 정성적 성능 평가

시나리오

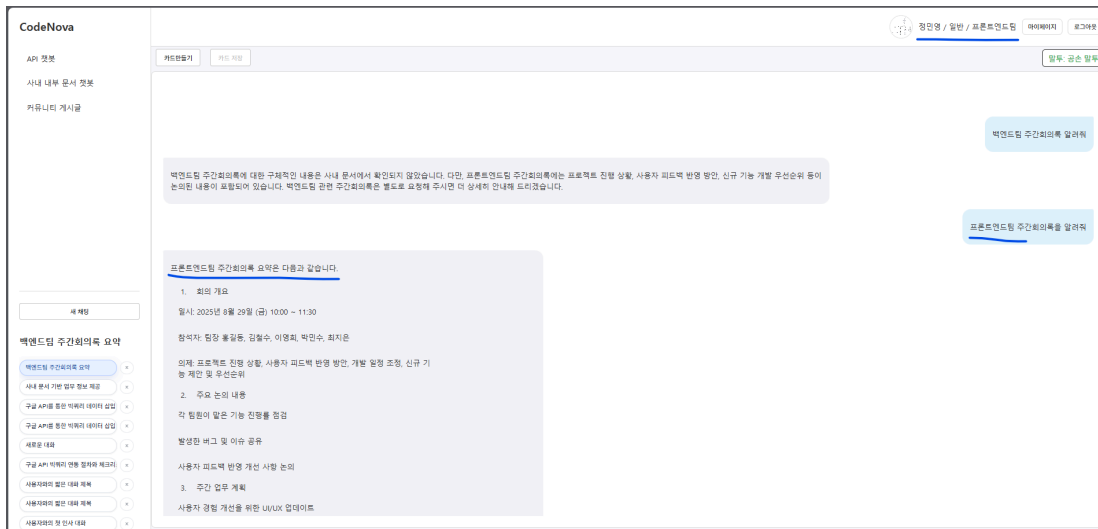
- ① 각 팀/권한에 맞는 내용 답변 여부 확인(권한 없는 문서 접근 불가)

테스트 결과:

- 프론트엔드팀으로 로그인 후, 백엔드팀 문서 질문 : 문서에 없는 내용으로 답변 못한다고 함



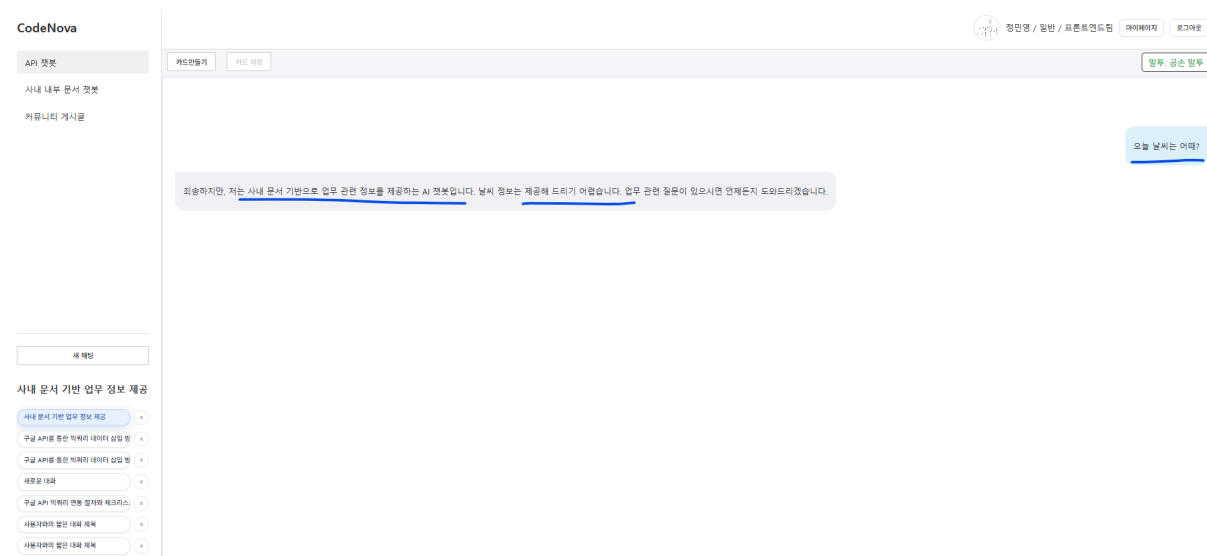
- 프론트엔드팀으로 로그인 후, 프론트엔드 팀 문서 질문 : 문서 내용 검색해서 답변해줌



② 문서에 없는 내용 질문 시 '검색 결과 없음' 응답 확인

테스트 결과:

- 날씨 질문: 답변 못한다고 함.

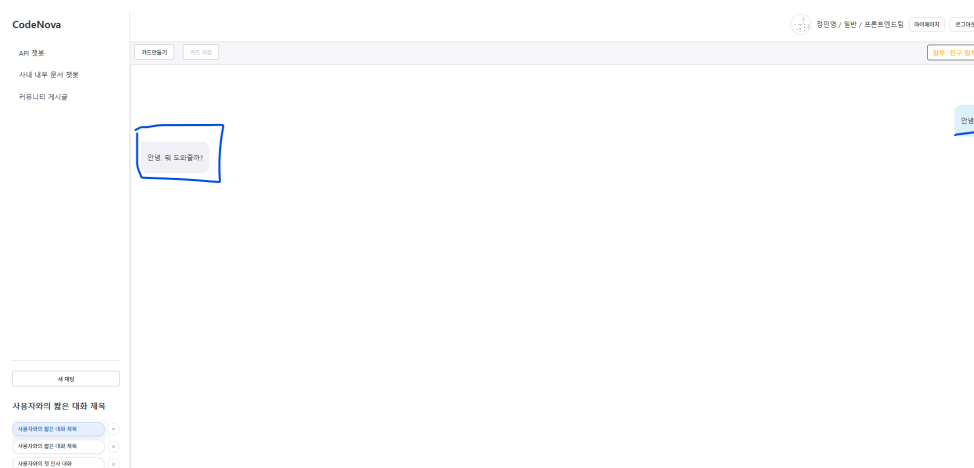


- 마케팅팀 주간회의록 질문: 문서에 없어서 답변 못한다고 함.



③ 일상 질문은 tool_call 없이 바로 답변 여부 확인

테스트 결과: tool_call 없이 바로 답변 (실제 로그에서도 tool 호출 없이 바로 답변)



```

1 response = chat({'history': [
2   {'role': 'user', 'content': '안녕'}, {'tone': 'formal', 'permission': 'cto', 'k': 8}]
3
✓ [o] 3s 158ms
  <tools>
  {'type': 'function', 'function': {'name': 'cto_search', 'description': '사내 문서 검색을 위한 도구입니다. 대화 내역로 바탕으로 사용자가 원하는 문서를 찾고, 관련된 문서를 반환합니다.', 'parameters': {'type': 'object', 'properties': {'keyword': {'type': 'string', 'description': '검색할 문서 키워드 (예: '코드노바 API 서버 설정')'}}}, 'required': ['keyword'], 'additionalProperties': false}}}
  </tools>

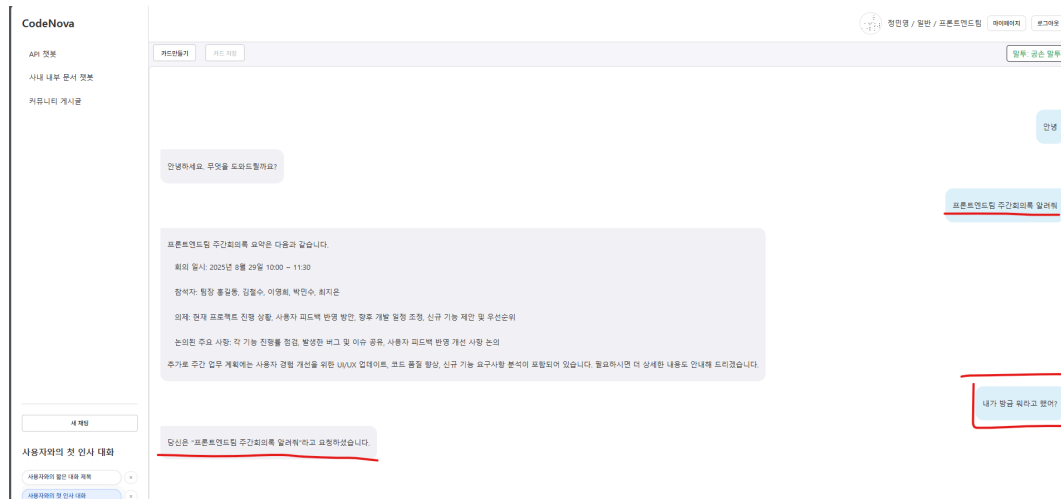
  For each function call, return a json object with function name and arguments within <tool_call></tool_call> XML tags:
  <tool_call>
  {'name': <function-name>, 'arguments': <args-json-object>}
  </tool_call>

INFO:httpx:HTTP Request: POST https://api.runpod.ai/v2/fcaeo5umuhd4vj/openai/v1/chat/completions "HTTP/1.1 200 OK"
INFO:utils_qwen3_topk.service:----- LLM Tool Parse Response Success
INFO:utils_qwen3_topk.service:----- LLM Tools Match : 0
INFO:utils_qwen3_topk.service:----- Final Assistant Reply generated
INFO:utils_qwen3_topk.service:----- 안녕하세요. 무엇을 도와드릴까요?

```

④ 멀티턴 대화 맥락 반영 여부 확인

테스트 결과: 이전에 물어본 내용 기억해서 답변



⑤ 선택된 말투에 맞게 답변 여부 확인

선택된 말투: 공손 말투

테스트 결과: 공손한 말투로 잘 대답해줌.



선택된 말투: 친구 말투

테스트 결과: 친근 말투로 잘 대답해줌.

