



팀 메추리알

RAG 기술을 활용한 리뷰 및 실시간 데이터 기반 맛집 추천 챗봇

Context-Aware Recommendation RAG Hybrid Retrieval Real-time

| 팀 소개 : 메추리알



메뉴(Menu)

메뉴 단위 데이터 집중



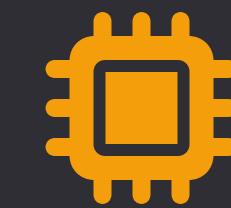
추천

사용자 맞춤 추천



리얼타임 (Real-time)

실시간 데이터 반영



알고리즘 (Algorithm)

고도화된 추천 로직

팀원: 김재은, 김현지, 한정화

| 목차

01 문제 정의 & 연구 목표

02 데이터 수집 & 전처리

03 연구 방법

04 연구결과

05 결론 및 기대효과

06 Q&A

| 문제 정의 & 연구 목표

기존 검색의 한계: 맥락의 부재



결정 피로

관련 없는 정보와 광고로 인한 피로감.



할루시네이션

기준 챗봇은 없는 식당을 추천하기도 함.



광고 포화

상단 노출 대부분이 광고성 게시글.



관련 없는 리뷰

내 상황(예: 조용한 회식 장소)에 맞는 리뷰를 찾기 어려움.

건대에서 회식하기 좋은 분위기·좌석·메뉴 기준으로 골라서 추천해줄게!
(단체 수용, 깔끔한 분위기, 회식 만족도 높은 곳 위주)

1. 고기·구이류 회식

① 연타발 건대점 (돼지고기·숙성구이)

- 특징: 깔끔한 실내, 직원이 구워줘서 편함
- 메뉴: 숙성삼겹·목살, 간장새우, 냉면
- 분위기: 너무 시끄럽지 않고 회사 회식으로 무난
- 추천 이유: 회식 만족도 높은 '안전한 선택'

② 우대갈비 건대점

- 특징: 고급스러운 프리미엄 갈비집
- 분위기: 단체석 가능 / 깔끔 & 조용
- 추천 이유: 예산 여유 있는 회식이면 강력 추천

2. 국물·한식 계열 회식

③ 황우양곰탕 건대점

- 특징: 든든한 한식 회식, 조용함
- 메뉴: 곰탕/수육/수육무침
- 추천 이유: 과음 없이 담백한 회식 원활 때 적합

④ 남도분식 (한식·해산물)

- 특징: 식사 + 가벼운 술 가능한 집
- 메뉴: 꼬막비빔밥, 낙지볶음, 제철 해산물

-> 전부 존재하지 않는 식당으로 할루시네이션 발생

연구 목표

★ 신뢰도 높은 추천

실제 리뷰 기반 신뢰도 높은 식당 메뉴 추천 모델 구축

↖ 맞춤형 추천

사용자의 개별 상황과 조건을 반영한 메뉴단위 추천

↳ 대화형 챗봇

대화형 챗봇형태로 제공하여 필요한 정보를 직관적으로 획득하게 함

| 데이터 수집

Selenium/공식 API 활용해 카카오맵과 구글맵에서 **서울숲역, 성수역, 건대입구역, 뚝섬역, 어린이대공원역 반경 2km** 안에서 음식점 **2,465개** 의 점포를 대상으로 약 **5만건** 의 리뷰 데이터와 **2만9천건** 의 식당정보 및 메뉴 정보를 수집

음식점 기본 정보

출처

카카오맵

데이터 수집방법

공식 API 수집 및 크롤링

대상

서울숲역, 성수역, 건대입구역, 뚝섬역, 어린이대공원 인근 음식점(카페 포함)

데이터 크기

2,465개 점포

수집 항목

상호명, 주소, 지하철역 출구정보, 업종, 요일별 영업시간, 메뉴목록, 가격, 메뉴 설명 등 총 51가지 항목

음식점 리뷰 데이터

출처

카카오맵, Google 지도

데이터 수집방법

Selenium BeautifulSoup 동적 크롤링

대상

기본 정보에서 수집한 점포

데이터 크기

총 49,903 건

수집 항목

상호명, 리뷰 텍스트, 평점, 리뷰어 정보, 리뷰 작성 일자, 출처

| 데이터 수집 : 서울시 실시간 데이터

맥락적이고 동적인 데이터를 활용하기 위해 **서울열린데이터광장 API**를 활용. 실시간 인구 혼잡도, 날씨, 주차장 정보를 활용하며, 사용자 질의가 들어오면 **API로 실시간 요청하여 반영**.



실시간 인구 밀도 정보를 제공하여 혼잡한 지역을 피한 추천 가능.



현재 날씨 정보를 반영하여 날씨에 맞는 메뉴 추천 가능.



Haversine 공식을 활용하여 가까운 주차장 정보를 실시간으로 제공.

```
<AREA_NM>성수카페거리</AREA_NM>
<AREA_CD>POI068</AREA_CD>
▼<LIVE_PPLTN_STTS>
  ▼<LIVE_PPLTN_STTS>
    <AREA_NM>성수카페거리</AREA_NM>
    <AREA_CD>POI068</AREA_CD>
    <AREA_CONGEST_LVL>약간 불편</AREA_CONGEST_LVL>
    <AREA_CONGEST_MSG>사람들이 몰려있을 가능성이 크고 불편다고 느낄 수 있어요. 인구밀도가 높은 구간에서는 도보 이동시 부딪침이 발생할 수 있어요..</AREA_CONGEST_MSG>
    <AREA_PPLTN_MIN>40000</AREA_PPLTN_MIN>
    <AREA_PPLTN_MAX>42000</AREA_PPLTN_MAX>
    <MALE_PPLTN_RATE>49.1</MALE_PPLTN_RATE>
    <FEMALE_PPLTN_RATE>50.9</FEMALE_PPLTN_RATE>
    <PPLTN_RATE_0>0.8</PPLTN_RATE_0>
    <PPLTN_RATE_10>6.0</PPLTN_RATE_10>
    <PPLTN_RATE_20>24.8</PPLTN_RATE_20>
    <PPLTN_RATE_30>25.3</PPLTN_RATE_30>
    <PPLTN_RATE_40>17.4</PPLTN_RATE_40>
    <PPLTN_RATE_50>14.0</PPLTN_RATE_50>
    <PPLTN_RATE_60>7.9</PPLTN_RATE_60>
    <PPLTN_RATE_70>3.9</PPLTN_RATE_70>
    <RESNT_PPLTN_RATE>50.0</RESNT_PPLTN_RATE>
    <NON_RESNT_PPLTN_RATE>50.0</NON_RESNT_PPLTN_RATE>
    <REPLACE_YN>N</REPLACE_YN>
    <PPLTN_TIME>2025-12-10 13:45</PPLTN_TIME>
    <FCST_YN>Y</FCST_YN>
  ▶<FCST_PPLTN>
    ...
    </FCST_PPLTN>
  </LIVE_PPLTN_STTS>
  </LIVE_PPLTN_STTS>
  ▶<ROAD_TRAFFIC_STTS>
    ...
    </ROAD_TRAFFIC_STTS>
  ▶<PRK_STTS>
    ▼<PRK_STTS>
      <PRK_NM>세종빌딩 주차장</PRK_NM>
      <PRK_CD>3085899</PRK_CD>
      <PRK_TYPE>BS</PRK_TYPE>
      <CPCTY>44</CPCTY>
      <CUR_PRK_CNT/>
      <CUR_PRK_TIME/>
      <CUR_PRK_YN>N</CUR_PRK_YN>
      <PAY_YN>Y</PAY_YN>
      <RATES>3000</RATES>
      <TIME_RATES>10</TIME_RATES>
      <ADD_RATES>3000</ADD_RATES>
      <ADD_TIME_RATES>10</ADD_TIME_RATES>
      <ADDRESS>성동구 성수동2가 317-23</ADDRESS>
      <ROAD_ADDR/>
      <LNG>127.057285</LNG>
      <LAT>37.54372</LAT>
    ▶<PRK_STTS>
```

| 데이터 전처리 : 리뷰 데이터 (1/3)

리뷰 데이터 전처리

카카오맵과 구글맵에서 수집한 **49,903건**의 리뷰 데이터에 대해
체계적인 전처리 과정을 진행.

전처리 과정

리뷰 데이터에 존재하는 **비속어, 축약어, 온라인 용어, 불용어, 이모지/광고 문구, 중복 리뷰**를 제거

1. Raw Crawling Data

카카오맵/Google 지도 리뷰

49,903 건



2. Deduplication

중복 제거

중복 리뷰 및 광고 문구 제거



3. Text Preprocessing

텍스트 정제

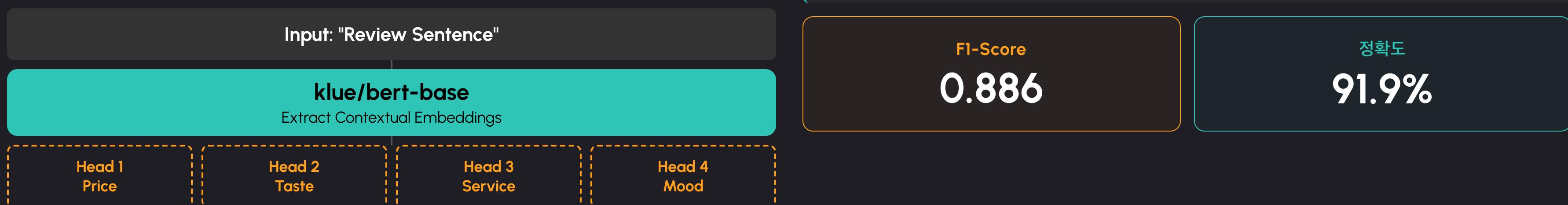
비속어·축약어·불용어·이모지 제거

| 데이터 전처리 : 리뷰 데이터 (2/3)

리뷰 감성분석

약 5만건의 리뷰데이터를 전부 사람이 레이블링하는 것은 현실적으로 불가능하여, **Silver Labeling** 기법을 도입. 1만 7천건의 리뷰데이터를 뽑아 **GPT-4o-mini**를 활용하여 각 리뷰를 **맛, 가격, 분위기, 서비스** 4가지 항목으로 각각에 대해 **긍정, 부정, 중립**의 감성 값을 레이블링(Silver labeling)하여 학습데이터를 구축.

ABSA(Aspect-Based Sentiment Analysis) 는 정보 추출과 감성 분류가 결합된 복합적인 작업으로, 리뷰 내 여러 속성(맛, 가격, 분위기, 서비스)을 개별적으로 식별하고 각각에 대한 감성을 분석하는 속성 기반 감성 분석 기법.	K-Fold 교차 검증 (3회) 데이터 불균형 해소를 위해 클래스 비율을 유지하며 3번의 교차 검증을 수행. 편차 없는 안정적인 성능이 나오도록 함.
--	---



▣ 리뷰 감성 분석 예시

리뷰 예시	가격	서비스	맛	분위기
"가성비는 별로... 양파닭이 너무 매웠음... 테이블 간격은 넓어서 좋았어요"	부정	중립	부정	긍정
"직원분들이 정말 친절하고 인테리어가 예뻐요. 비싸지만 가치 있어요."	중립	긍정	중립	긍정

이 모델을 통해 도출된 '**맛, 가격, 분위기, 서비스**' 4개 항목의 감성 점수는 **식당별 정량 지표**로 변환되어, **RAG 시스템 1차 필터링 및 메뉴 기반추천** 순위에 사용 활용됨.

| 데이터 전처리 : 리뷰 데이터 (3/3)

리뷰 임베딩

5만 개 리뷰 데이터를 AI가 검색 가능한 방식으로 만들기 위해, 크롤링한 리뷰 데이터를 리뷰 단위로 Split 함.

또한 분할된 텍스트를 AI가 이해할 수 있는 숫자 벡터로 임베딩하기 위해 **BM-K/KoSimCSE-roberta-multitask** 모델을 사용.

1

Split & Chunking

각 리뷰 단위로 구분

2

Embedding

BM-K/KoSimCSE
roberta-multitask

3

Vector DB 저장

Chroma

최적화 : Vector DB에는 수집한 전체 정보를 저장하지 않고 Retriever 사용에 필수적인 리뷰 정보만 별도로 구분해서 저장함으로써 RAG 속도를 빠르게 하고자함.

*KoSimCSE-RoBERTa는 기존 한국어 BERT·RoBERTa보다 '문장을 의미적으로 잘 구분·군집하도록' 대비 학습(SimCSE 방식)을 추가로 수행해 문장 임베딩 품질이 훨씬 뛰어남

| 데이터 전처리 : 메뉴 데이터

카카오맵에서 수집한 약 2만9천 건의 식당정보 및 메뉴 정보를 수집

✖ 1. 메뉴명 토크나이징

메뉴 이름에 따른 string matching을 위해 토크나이징 수행.

철판새우볶음밥
새우 볶음밥 철판

❤ 2. 메뉴 속성

메뉴 이름으로 추론될 수 있는 각 메뉴에 대한 속성을 구분하여 15개의 태그를 생성 (Gold Labeling).

카테고리

- 조리 형태: 국물, 면, 밥, 베이커리
- 맛과 온도: 차가움, 따뜻함, 매움
- 건강 식단: 비건, 다이어트, 디저트, 건강식
- 재료: 고기, 해산물
- 주류 페어링: 주류, 음료

▣ 메뉴 데이터 예시

메뉴명	토큰	따뜻함	밥	해산물	매움
철판새우볶음밥	['새우', '볶음밥', '철판']	✓	✓	✓	✗
매운해물짬뽕	['매운', '해물', '짬뽕']	✓	✗	✓	✓

〽 3. 식당별 평점 값

리뷰 수가 적은 식당의 평점을 왜곡을 보정하기 위해 베이지안 평균을 적용하여 평점을 보정합니다.

Bayesian average

- 리뷰 수가 적은 식당의 평점을 전체 평균으로 보정
- 식당별 정량 지표로 활용하여 추천 시스템에 적용

| 연구방법

• 사용자(User) 입력 자연어 질의(Query)



Q 1) 이중 검색 Dual Retrieval

① 속성 가중치 & 메뉴명 기반 문자열 매칭

(Attribute Weight & String Matching)

속성/시간대 가중치 적용 / 명확한 메뉴 키워드 처리 / 식당 및 메뉴 각 2곳 후보 선정

② 리뷰 임베딩 기반 의미 검색

(Embedding Similarity)

서술형 리뷰 의미 분석 / 식당 3곳 후보 선정



후보 식당 리스트 통합 (메뉴 기반 4곳 + 리뷰 기반 3곳)



2) 내부 데이터 수집

사전 저장 META DATA 조회 (식당명, 주소, 메뉴, 리뷰 수 등)

3) 외부 데이터 수집

서울시 실시간 API 호출 (혼잡도, 영업시간 등) → JSON 변환



4) LLM 정보 종합 처리

모델 : GPT-4o-mini

내부 META DATA + 외부 실시간 데이터 동시 입력(JSON 형식) / 프롬프팅에 따른 추론과정 진행(chain of thought) / LLM 종합 판단으로 최종 결과 생성



• 사용자에게 최종 답변

추천 식당 + 추천 이유 + 조건 적합성 + 실시간 정보

| 연구방법 : 메뉴 속성 가중치

사용자 질문에서 Okt 형태소분석기를 활용하여 토크나이징한 후 핵심 요소를 추출하여 정형화된 딕셔너리 값으로 구조화합니다.

질의 예: "성수역에서 저녁으로 매운 국물이 먹고 싶어"

1. 지역 필터링

[성수]
29,051 rows →
6,324 rows

속성 기반

2. 속성
시간 가중치
적합도 점수

메뉴명 기반

2. 토큰-메뉴명
String Matching

점수 결합

속성/시간 80%
평점 20%

4. 최종 순위 정렬

점수 순 정렬 · 동점 시 맛 점수로 순위 결정

속성 + 시간대 가중치 적용

감지된 속성 → 매운(spicy), 국물(soup)

시간대 → dinner → 매운/국물/술·고기 페어링 가중치 ↑

가중치 적용 후 비교

식당명	메뉴명	속성 속성 80%	시간대 속성+시간대 80%	평점 평점 20%
보끔당 성수점	보끔당 닭볶음탕	1.725670	2.075670	1.375670
갓벌의조개 성수역점	대왕조개전골 小	1.722661	2.072661	1.372661
우리칼국수	동죽칼국수	1.649419	1.999419	1.299419
진지방	순대국	1.633644	1.983644	1.283644
조조칼국수 성수점	동죽칼국수	1.624150	1.974150	1.274150

순위 변화

보끔당 성수점
9위 → 1위 ↑ +8
갓벌의조개 성수역점
10위 → 2위 ↑ +8
우리칼국수
11위 → 3위 ↑ +8
진지방
12위 → 4위 ↑ +8
조조칼국수 성수점
14위 → 5위 ↑ +9

최종 점수 결합

80% 속성 적합도 + 20% 베이지안 평점

최종 추천 결과

1위: 보끔당 성수점 - 보끔당 닭볶음탕

2위: 갓벌의조개 성수역점 - 대왕조개전골 小

| 연구방법 : 메뉴 String Matching

사용자 질문에서 Okt 형태소분석기를 활용하여 토크나이징한 후 핵심 요소를 추출하여 정형화된 딕셔너리 값으로 구조화합니다.

질의 예: "뚝섬역 근처에 아이스크림이 맛있는 카페"

1. 지역 필터링

["뚝섬"]

29,051 rows →

3,551 rows

◆ 속성 기반

2. 속성/시간 가중치

적합도 점수

메뉴명 기반

2. 토큰-메뉴명 String Matching

3. 점수 결합

속성/시간 80%
평점 20%

3. 점수 결합

키워드 매칭 80%
평점 20%

4. 최종 순위 정렬

점수 순 정렬 · 동점 시 맛 점수로 순위 결정

메뉴명 키워드 string matching

키워드 매칭 → 아이스크림, 카페

키워드 매칭 점수 비교

식당명	메뉴명	토큰값	키워드 점수 키워드 점수	키워드 매칭 키워드매칭 80%	평점 평점 20%
호미담	집 가기 전 아이스크림	집, 가기, 아이스크림, 전	1.0	1.625212	1.275212
피비지랩	아이스크림	아이스크림	1.0	1.611047	1.261047
샌드보이	카페라떼	카페라떼	0.7	1.564703	1.319703
멜로우즈	소금빵아이스크림	소금, 아이스크림, 빵	1.0	1.564059	1.214059
세븐버거스 어 위크	카페라떼	카페라떼	0.7	1.528285	1.283285

최종 점수 결합

80% 키워드 매칭 점수 + 20% 베이지안 평점

🏆 최종 추천 결과

1위: 호미담 - 집 가기 전 아이스크림

2위: 피비지랩 - 아이스크림

| 연구방법 : RAG (1/2)

1차 필터링 – Sparse Retrieval

⚠ 문제점
질문이 들어올 때마다 5만 건의 리뷰 전체를 대상으로 유사도 검색
→ 입력 길이 증가 및 높은 지연 시간

💡 해결책: Sparse Retrieval
사전에 메타데이터로 저장해둔 식당별 역·시간대·카테고리·속성 4가지 항목으로 pre-filtering 수행

💬 예시 쿼리
"성수역에서 저녁에 가면 좋을 분위기 괜찮은 양식당 추천해줘"

성수 저녁 분위기 양식당

필터링 과정 예시

역 필터링
성수역 인근
46,858건
→ 14,384건
69.3% 감소

시간대 필터링
저녁 영업
14,384건
→ 13,102건
8.9% 감소

카테고리 필터링
양식당
13,102건
→ 2,484건
81.0% 감소

속성 필터링
분위기 ≥ 0.7
2,484건
→ 2,260건
9.0% 감소

최종 결과

46,858건 → 2,260건

52개 식당의 리뷰 · Vector DB 임베딩 로드: 2,260건

검색 범위 95% 감소 · 소요 시간 대폭 개선

| 연구방법 : RAG (2/2)

2차 필터링 - 리뷰 임베딩 기반 Dense Retrieval

1차 필터링에서 걸러진 식당의 리뷰를 대상으로 Dense Retrieval 기반의 코사인 유사도를 활용해 쿼리와 리뷰의 의미적 유사도를 계산. 세 가지 방식을 적용한 뒤, 가장 성능이 우수한 방법을 최종적으로 선정.

① 기본 유사도

각 리뷰의 코사인 유사도를 계산한 뒤 식당별로 상위 10개의 평균값을 구하여 평균 유사도가 높은 식당을 선택하는 방법

② Time Weighted

기본 유사도 방식에 따라 나온 결과에 더해 리뷰 작성일자를 고려해 최근 리뷰에는 높은 가중치를, 오래된 리뷰에는 낮은 가중치를 부여하도록 추가 반영

③ MMR

쿼리와의 유사도는 높게 유지하면서도 이미 선택된 리뷰와의 중복을 줄여 정보의 다양성과 정확성을 동시에 확보하도록 함

K를 3으로 설정, 각 방법마다 3개의 식당을 선정, 다음과 같은 정확도 판정결과를 도출

정량적 평가

1) 기본 유사도 방식

0.87

2) Time Weighted

0.89

3) MMR

0.81

정성적 평가

1. 성수 디저트 맛집 추천해줘

순위	기본 Retrieve	Time Weighted	MMR
1위	파리바게트	양과자점 라보엠	양과자점 라보엠
2위	스노우베리	샹도르	샹도르
3위	-	에르제 성수	아이토아이
정답 개수	2	3	3

총 40건의 예시 쿼리를 바탕으로 성능평가 비교

| 연구방법 : 실시간 데이터 적용

추천의 현실성을 완성하기 위해 서울시 열린데이터광장의 API를 연동해 실시간 데이터를 적용.



주차장 데이터

추천 식당이나 지하철 역을 기준으로
하버사인 공식을 적용하여 정밀한 거리를 계산.

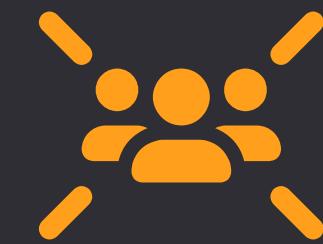
실시간 주차 가능한 대수를 제공하여
차량 이용자의 편의를 높임
(예: '성수명당'에서 가장 가까운 주차장은
서울숲 공영주차장, 252m)



실시간 기상정보

메뉴 속성 데이터나 리뷰 데이터와 결합하여
상세하게 식당과 메뉴를 추천하도록 했음.

(예: 비 오는 날 따뜻한 국물, 더운 날 시원한 면)



혼잡도 데이터

실시간 혼잡도 상황을 제공하여
방문 전에 참고할 수 있도록 하였음.
(붐비는 시간대 회피 또는 활기찬 분위기 선호 시 활용)

| 연구방법 : LLM 프롬프트 설정

LLM 프롬프트 설계

프롬프트 설계 단계에서는 모델이 상황에 맞는 답변을 생성할 수 있도록 여러 구성 요소를 체계적으로 활용.

① System Message, Human Message, AI Message 대화 구조

System Message는 모델의 역할과 페르소나를 정의하여 일관된 톤과 방식으로 응답하도록 지시하고, Human Message는 사용자가 입력하는 실제 요구사항을 전달하며, AI Message는 모델이 생성하는 최종 출력 내용을 의미.

② ChatPromptTemplate & MessagesPlaceholder

ChatPromptTemplate과 MessagesPlaceholder를 사용해 챗봇의 대화 흐름을 설계. 이를 통해 모델이 대화 맥락을 유지하면서 질문 의도에 맞는 답변을 추론하여 생성할 수 있도록 함.

③ Few-shot 기법

실제 사례를 제공하여 답변 패턴을 학습시키는 방식으로, 여러 연구에서도 few-shot prompting이 모델의 추론 정확도와 일관성을 개선하는 데 효과적이라는 결과가 보고된 바 있음.
실시간 API를 이용해 인근 주차장에 대한 정보를 가져올 때 주차에 해당하는 KEY값을 활용하는 답변 방법을 알려줌.

④ ChatMessage History

대화형 RAG 시스템에서는 과거 대화 이력도 중요한 정보가 되기 때문에 ChatMessage History 기능을 활용하여 이전 대화를 저장하고, 이를 답변 생성에 반영할 수 있도록 구성.

⑤ LCEL (LangChain Expression Language)

LCEL을 활용해 프롬프트, 모델 호출, 응답 후처리 등 전체 파이프라인을 하나의 체인처럼 유기적으로 연결. 이를 통해 프롬프트 구성, 모델 추론, 후처리 단계가 일관된 구조로 실행될 수 있도록 했음.

① System Message, Human Message, AI Message

```
System: "당신은 음식점 추천 챗봇입니다..."  
Human: "성수역에서 맵네 국물이 먹고 싶어"  
AI: "보끔당 성수점을 추천드립니다..."
```

② ChatPromptTemplate & MessagesPlaceholder

```
prompt = ChatPromptTemplate.from_messages([  
    ("system", "..."),  
    MessagesPlaceholder("history"),  
    ("human", "{input}")  
])
```

③ Few-shot 기법

```
실시간 API 호출 시:  
KEY: "PRK_NM" → 성수 1 공영 주차장
```

④ ChatMessage History

```
사용자: "보끔당 성수점 추천해줘"  
AI: "보끔당 성수점을 추천드립니다..."  
  
사용자: "그 식당 주변 주차 정보 알려줘"  
→ 이전 대화에서 "보끔당 성수점" 참조
```

⑤ LCEL (LangChain Expression Language)

```
chain = prompt | llm | output_parser  
→ 프롬프트 → 모델 → 후처리 연결
```

| 연구결과

정확도 비교

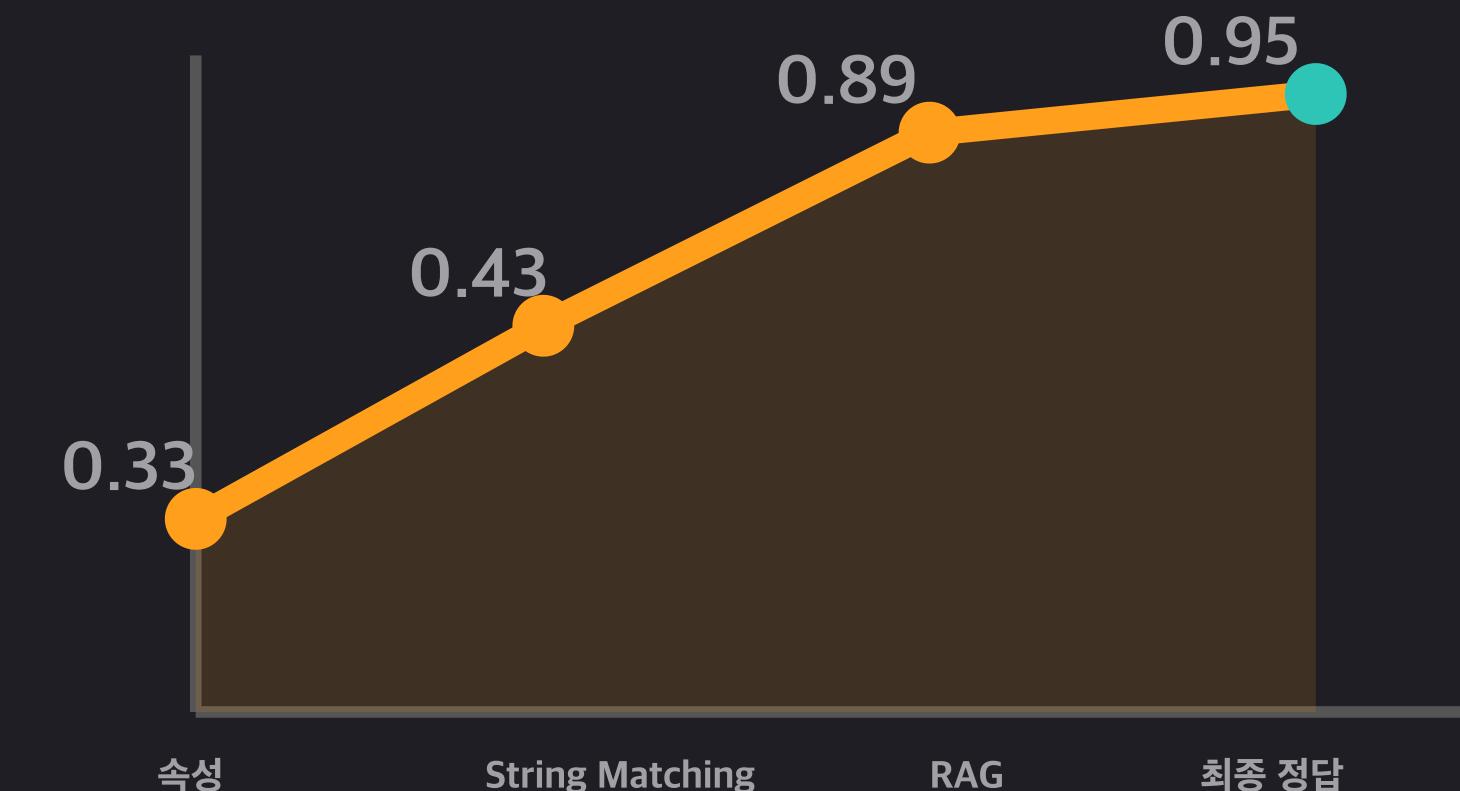
듀얼 방식과 프롬프팅을 적용한 결과 최종 정답율 0.95의 결과를 도출.

메뉴 속성 **0.33 (33%)**

메뉴 스트링매칭 **0.43 (43%)**

Time Weighted Retrieval **0.89 (89%)**

최종 정답 **0.95 (95%)**



최종 정답률 0.95 달성.

메뉴명 string matching 및 메뉴 속성 가중치 + RAG를 결합한 듀얼 방식과 프롬프팅을 통해 최종 정답율 0.95를 달성.

| 데모 시연

예시 1

"성수에서 졸업축하파티를 하려는데 인테리어 예쁜 맛집 추천해줘"

예시 2: 후속 질문

- 1) 첫번째 식당 근처에 주차장이 있어?
- 2) 오밀 성수 식당에 현재 날씨에 어울리는 메뉴 알려줘
- 3) 지금까지 추천 식당들 메뉴 가격대가 어때?
- 4) 이 중에서 예약 가능한 곳은? 사람많아?
- 5) 몇 시까지 영업해?



동영상 녹화본 클릭
클릭하여 재생

| 결론 및 기대효과

💡 프로젝트 의의

본 프로젝트의 가장 큰 의의는 기존 맛집 검색의 패러다임을 단순 키워드 매칭에서 맥락기반 추천으로 전환했다는 점.

✓ 주요 성과

- 기존 검색이 잡아내지 못했던 "분위기 좋은", "비 오는 날 국물"과 같은 추상적인 니즈의 속성 분석을 통해 구체적인 검색 필터로 변환
- LLM의 고질적인 문제인 할루시네이션을 약 5만건의 실제 리뷰 데이터에 근거한 RAG 시스템으로 보정하여 검증된 사실 기반의 추천 구현

⚠️ 한계점

철저히 리뷰 데이터에 기반하기 때문에 리뷰에 언급되지 않은 정보나, 희소한 키워드에 대해서 답변을 하지 못하는 경향 존재. 이는 할루시네이션을 막기 위한 트레이드 오프였으나, 향후 데이터 수집 지역을 확장하고 메타데이터 범위를 넓힌다면 충분히 해결 가능할 것으로 판단됨.

👉 향후 과제

현재는 단발성 추천에 그치지만 사용자 ID기반의 데이터 축적과, 강화학습을 적용한다면, 사용할수록 내 취향을 더 잘 알고 파악하는 초개인화 추천 엔진으로 고도화 할 수 있을 것으로 기대됨.

"메추리알은 허기를 '근거'로 달래드립니다."



Q&A

질문을 주시면 성심껏 답변드리겠습니다