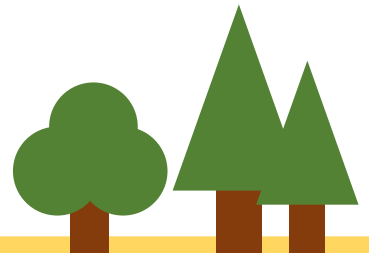
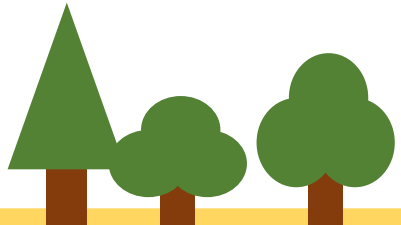


머먹찌

인공지능 기반의 배달추천 챗봇 서비스

iLifO 8층_맛집 = ['강가영', '김관훈', '박현준', '임도연']



1. 프로젝트 개요

- 팀 소개
- 챗봇 현황
- 프로젝트 목표

2. 데이터 수집 및 전처리

- 데이터 수집
- 크롤링
- 데이터 전처리
- 추천 시스템 자료 수집

3. 개발 프로세스

- 챗봇 구성도
- 챗봇 시나리오 설계
- 챗봇
- 기술 적용
- GPT2

4. 시연

- 추천 엔진
- koGPT2

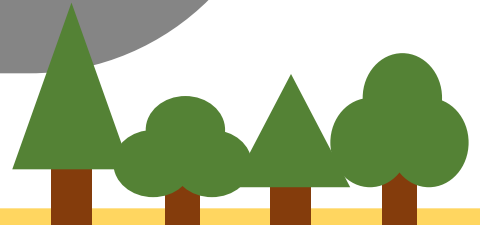
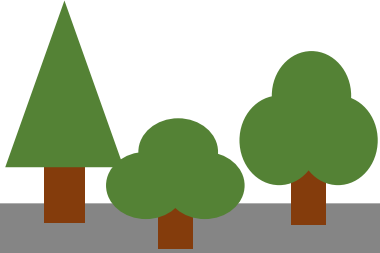
5. 기대효과

6. 향후 계획

- 보완 및 수정
- 서버 연결

7. 개발 환경

1. 프로젝트 개요



1. 프로젝트 개요

산학 프로젝트

- 기간 : 3개월(8월 3일 ~ 11월 6일)
- 구성 : 기획 및 분석 - 설계 - 개발 - 테스트 - 유지보수
- 멘토 : 김복주 PE
- 멘티 : 강가영, 김관훈, 박현준, 임도연

챗봇

- 주요 기능 : 배달 맛집 추천, 질의응답, 맛집 검색, 문맥 이해, 감정 및 질문 의도 분류
- 구현 : 맛집 추천봇 머먹찌 시연 영상
- 기대 효과 : 로컬 맛집을 추천 받고자 하는 사용자 의도 파악한 답변 기능 제공

1. 프로젝트 개요

팀 구성원 및 R&R



Sw Architecture,
DB, Server 설계 지도
Chatbot Engine 관련
Deep-Learning 지도
Machine Learning
후속 연계 기술전수



역할 : 데이터 수집,
RDB 설계,
Chatbot Engine 개발,
맛집 추천 Engine 개발,
koGPT2 적용



역할 : 팀장,
데이터 수집,
Chatbot Engine 개발,
맛집 추천 Engine 개발,
koGPT2 적용



역할 : 데이터 수집,
Chatbot Engine 개발,
맛집 추천 Engine 개발,
koGPT2 적용

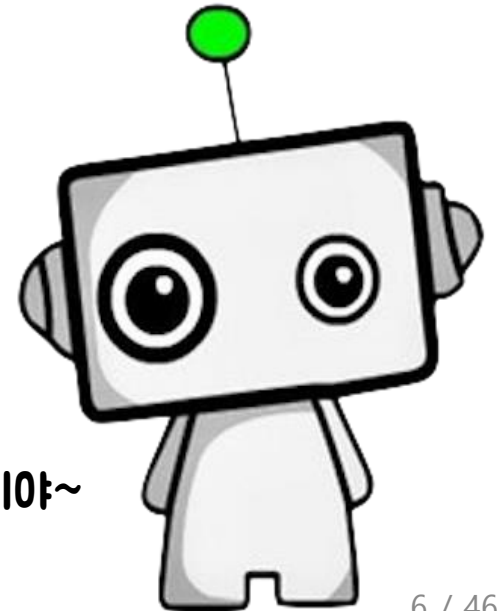


역할 : 데이터 수집,
GraphDB 설계,
Chatbot Engine 개발,
맛집 추천 Engine 개발,
koGPT2 적용

챗봇이란?

챗봇은 음성이나 문자를 통한 인간과의 대화를 통해서
특정한 작업을 수행하도록 제작된 컴퓨터 프로그램입니다.

안녕!
나는 챗봇이야~



1. 프로젝트 개요

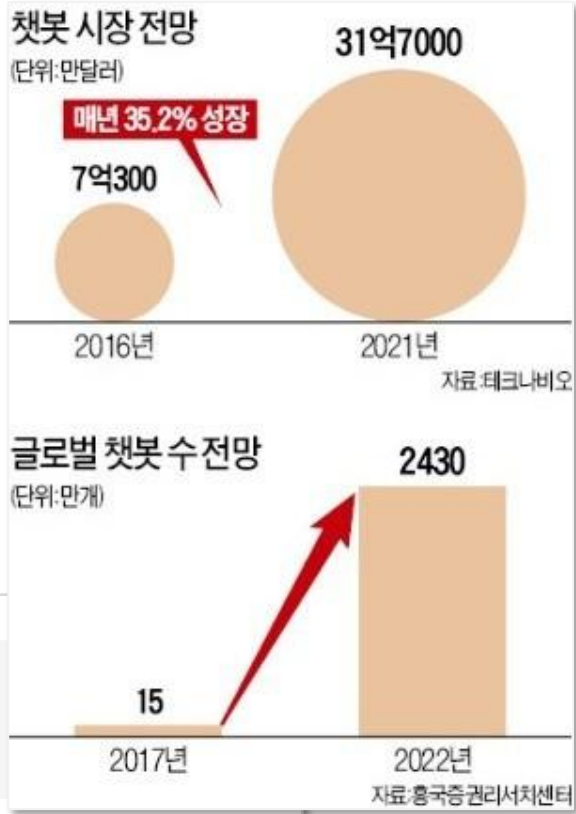
챗봇 현황

챗봇 트렌드 2020: 1인 1봇 시대의 등장

머지않아 우리는 1인 1봇의 시대를 경험하게 될 것입니다. 많은 상점들은 점점 주문을 받기 위한 챗봇을 도입하고 있습니다. 현재는 프랜차이즈들 중심으로 키오스크와 함께 주문 챗봇을 도입하고 있습니다. 프랜차이즈들 중심으로 가고 있는 이유는 높은 비용문제 때문인데, 챗봇 빌더들이 사용하기 쉽게나오고 다양한 서비스들이 생기면서 점차 소규모의 상점들까지도 챗봇을 도입할 수 있게 되었습니다.



고객은 메신저를 통해서 커피주문을 할 수 있다.
(노컷뉴스 자료 참고, "카톡으로 커피주문?"...카카오, '챗봇주문' 서비스 출시)
A customer can order the coffee via 'Chatbot' on the messenger platform.



"2020년에도 챗봇 시장의 절대강자는 없다"

by Tony Around

방금 언급했듯이 시장의 팽창과 함께 수 많은 업체들이 챗봇 빌더(개발 플랫폼)를 출시하였습니다. 그리고, 각 자의 독자적인 기술들을 통하여 상호간에 도움을 주면서 발전해왔습니다.

하지만, 아직까지 절대적으로 시장을 지배하는 기업은 없습니다. 이유는 기업들이 챗봇을 구축하면서 현재까지도 실패를 경험하고 있고 인공지능 기술에 대한 정답이 없기 때문입니다. 특정 비즈니스를 위한 인공지능은 일부 있지만, 어딜가나 적용할 수 있는 공통의 인공지능 기술 혹은 상용화된 인공지능 기술이 없기 때문입니다.

1. 프로젝트 개요

프로젝트 배경 및 목표



머 먹 찌

머먹찌는 '오늘 뭐 먹지?' 라는 고민을 하는 여러 사람들에게
메뉴와 맛집을 추천해주는 챗봇입니다.

맛집, 메뉴, 가격, 평판 데이터를 이용하여 사용자가 원하는 정보를
즉각적으로 제공하는 것을 넘어서 날씨와 사용자의 기분, 위치 등을 토대로
맛집을 추천하는 초개인화 맞춤형 맛집 추천 챗봇입니다.

1. 프로젝트 개요

프로젝트 배경 및 목표

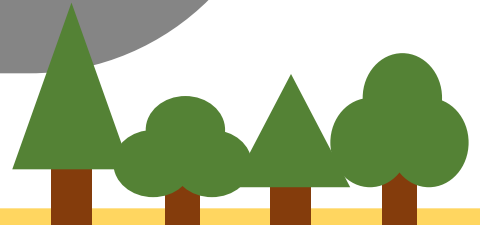
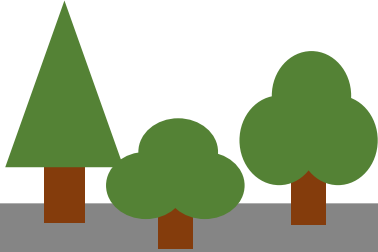
- ✓ 지역 맛집, 메뉴, 가격, 평판 정보 수집 및 데이터베이스 구축
- ✓ 공공데이터, 날씨, 지도 open API 연계 개발
- ✓ 인공지능 챗봇 서비스 엔진 개발 및 챗봇 UI 메신저 개발
- ✓ 인공지능 초개인화 추천서비스 엔진 개발

1. 프로젝트 개요

수행 계획

8월	8월	9월	10월	10월
계획	데이터 분석	설계	개발	테스트 및 유지보수
프로젝트 계획 수립 착수보고진행	유튜브 댓글,자막, 유튜버 정보 네이버 블로그 내용 맛집 포털 상세 정보 크롤링 및 전처리	아키텍처 설계 챗봇진행절차 설계 DB 설계	자연어 처리(NLP) 자연어 이해(NLU) 대화 관리(DM) 답변 생성(NLG) DB 구축 부가 기능 추가	DB 테스트 통합 테스트 오류 수정

2. 데이터 수집 및 전처리



2. 데이터 수집 및 전처리

데이터 수집

 DININGCODE

M A N G O
P L A T E



데이터 수집


HOME > 나혼자산다 > 오늘 뭐 먹지?

[이 앱 어때?] 망고플레이트VS다이닝코드, 혼밥 맛집 찾을 수 있는 어플

👤 허진영 | ⌚ 승인 2020.09.23 18:00 | 💬 댓글 0



앱 대표 이미지 (사진=구글플레이스토어)



조용히 자기만의 시간을 가지며 밥을 먹는 혼밥족들이 늘어나면서 혼밥집을 찾는 사람들도 많아졌다. 사람이 많은 가게에서 혼자 먹기란 쉽지 않다. 그렇다고 대중먹고 싶지도 않은 혼밥족들이 맛있는 맛집을 찾으려면 어떤 어플을 사용해야 할까? 나만의 혼밥 맛집을 찾을 수 있는 앱 망고플레이트와 다이닝코드를 비교해봤다.

인기기사

1

[그것이 궁금] 무도 정준하 아이돌로 바꾼 어...

2

[혼족 레시피] 에어프라이어로 만드는 '다이...

3

[그것이 궁금] 셀럽 남녀 다이어트 비법...아...

4

[1인가구 단산] 제주도 장년층 1인가구 260...

5

[영화추천] 집콕 할로윈, 넷플릭스 좀바크리...

6

[1인가구 단산] 코로나19 위기가구 긴급생계...

7

SK그룹 8개사, 한국 최초로 RE100 가입..환경...

8

[솔로이코노미 동향] CU·배달의민족·쌍성 外

9

[술직체험기] 19세 여행작가 된 술구, 스마트...

10

[금융단신] NH농협은행·BK기업은행 한화손...

 KB 금융그룹

2. 데이터 수집 및 전처리

크롤링

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	대분류	중분류	상호ID	이름	주소	위도	경도	전화번호	메뉴/가격	가격대	영업시간	정보	특징	평점	리뷰
37625	중식	회귀	S0037624	소성공자	서울 강남	37.51768	127.0212	02-516-88	18,900원	7000원	매일 11:30	단체석, 주차, 무선 인터넷, 남/녀 화장실 구분			
37626	중식	회귀	S0037625	천유향	서울 은평	37.61977	126.9191	010-5534-	변동대표	16000원		단체석, 포장, 배달, 예약, 무선 인터넷, 남/녀			
37627	중식	회귀	S0037626	북경마라탕	경기 고양	37.68591	126.7721	031-975-8	2,500원대	2500원	화요일 휴	단체석, 포장, 예약, 무선 인터넷			
37628	중식	회귀	S0037627	회귀마을	경기 오산	37.1589	127.0601	031-373-6	15,800원	15800원	목요일 12:	단체석, 주차, 예약, 무선 인터넷			
37629	중식	회귀	S0037628	대흥회귀	서울 양천	37.5266	126.8628	070-7576-	15,800원	8900원	매일 11:00	단체석, 예약, 무선 인터넷, 남/녀 화장실 구분			
37630	중식	회귀	S0037629	미가양꼬치	서울 강북	37.61198	127.0313	02-945-96	18,000원	10000원		단체석, 예약			
37631	중식	회귀	S0037630	얼땅자	서울 광진	37.53878	127.0664	02-467-28	12,000원	12000원	매일 12:00	단체석, 예약, 남/녀 화장실 구분			
37632	중식	회귀	S0037631	초원샤브	인천 부평	37.49809	126.7212	032-442-6	15,000원	7000원	평일 11:00	단체석, 포장, 예약, 무선 인터넷			
37633	중식	회귀	S0037632	불리아 흥	서울 마포	37.55659	126.922	02-335-66	21,000원	21000원	매일 11:00	단체석, 주차, 예약			
37634	중식	회귀	S0037633	천미천회	서울 광진	37.55649	127.0799	02-429-38	16,900원	12900원	매일 11:00 - 23:00	L.O. 22:00			
37635	중식	회귀	S0037634	아이상회	서울 마포	37.55284	126.9215	02-336-25	12,600원	12600원	매일 11:30	단체석, 포장, 예약, 무선 인터넷, 남/녀 화장실			
37636	중식	회귀	S0037635	베이징 회	서울 송파	37.51067	127.085	02-415-88	변동2인세	None	매일 12:00	단체석, 예약			
37637	중식	회귀	S0037636	삼국지	서울 마포	37.56096	126.924	02-322-89	18,000원	12000원	매일 11:30	단체석	2TV생생정보 339회	17.05.30. 삼색	

맛집 정보 수집

424	J00423	가리아게
425	J00424	쭈구미무침
426	J00425	죽
427	J00426	러시아
428	J00427	맥앤치즈
429	J00428	브리또
430	J00429	퓨전술집
431	J00430	한식 주점

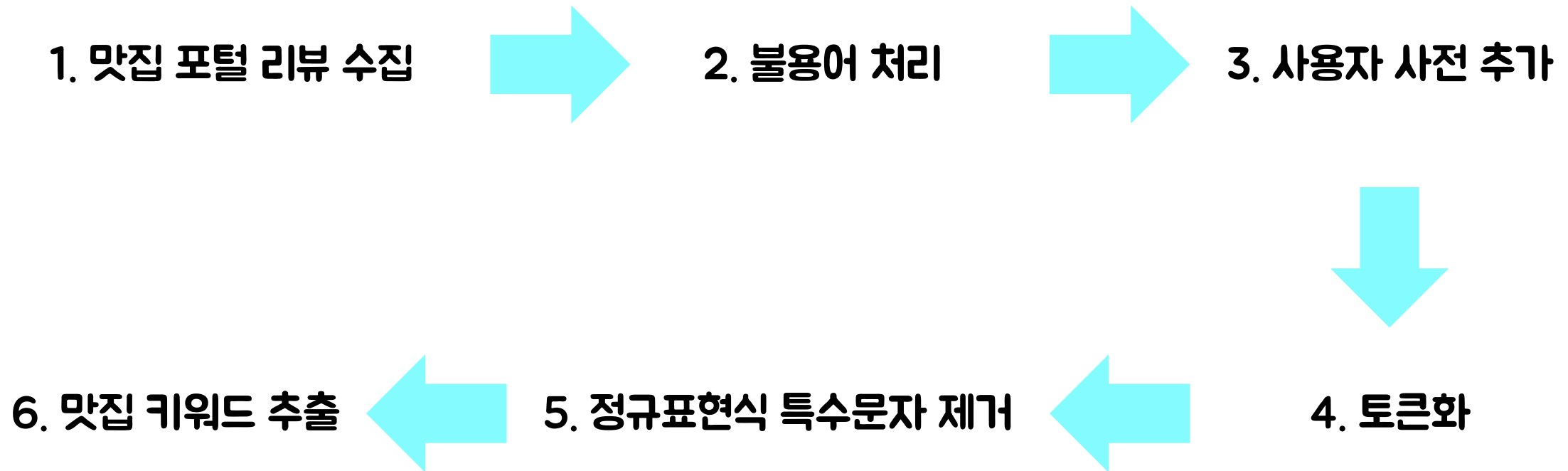
메뉴 리스트

37,637개의 상호명

431개의 메뉴

2. 데이터 수집 및 전처리

데이터 전처리



데이터 전처리 >

1) 리뷰 수집

```
import pandas as pd
df=pd.read_csv('1001네이버+망고+다이닝최종본.csv')
df['리뷰'].head(50)
```

0 더보기

1 1건의 방문자 평가4.0점맛5.0 가격5.0 서비스5.0방문목적점심식사(1)분위기캐...

2 3건의 방문자 평가5.0점맛5.0 가격5.0 서비스5.0방문목적식사모임(1) 기념일...

3 블로그리뷰사당_나인피스그릴 → 삼겹살에 버터 조합이라니언제나 먹어도 맛있는 삼겹살!...

4 1건의 방문자 평가5.0점맛5.0 가격5.0 서비스5.0방문목적회식(1) 술모임(1...

5 NaN

6 이 식당의 첫 평가를 작성해 주세요다녀온 맛집을 평가하면 다른 사용자에게 많은 도움...

7 1건의 방문자 평가5.0점맛5.0 가격3.0 서비스5.0방문목적저녁식사(1) 술모임...

8 블로그리뷰부산 현지인 로컬 맛집 : 남천동 촌식이네포차안녕하세요 여러분!! 부산 현...

9 NaN

10 더보기

11 1건의 방문자 평가4.0점맛5.0 가격3.0 서비스3.0방문목적술모임(1) 저녁식사...

12 NaN

13 연관맛집 재검색

14 이 식당의 첫 평가를 작성해 주세요다녀온 맛집을 평가하면 다른 사용자에게 많은 도움...

15 연관맛집육도가인계동 2.3km한우, 특수부위 삼익LA갈비인계동 2.5km1a갈비, ...

16 이 식당의 첫 평가를 작성해 주세요다녀온 맛집을 평가하면 다른 사용자에게 많은 도움...

17 연관맛집삼익LA갈비인계동 2.8km1a갈비, 계란말이 10번가의고깃집화성 5.2km...

18 연관맛집갈비원일산 3.5km갈비, 한정식 경복궁녹번 15.4km한정식, 생갈비 진진...

19 블로그리뷰[일산술집] 백석동 분위기좋은감성주점 운도운도 분위기 좋은 감성술집 백석역...

20 블로그리뷰돌갈비 무뎠나~~데이트팝 제휴광화문석갈비 솔직후기오늘은 광화문 석갈비를 먹...

21 블로그리뷰[200729] 수원 인계동 핫플술집: 어바웃#일상 #수원 #수원맛집 #수...

22 2건의 방문자 평가3.0점맛5.0 가격3.0 서비스5.0방문목적저녁식사(1)분위기푸...

23 1건의 방문자 평가0.0점맛0 가격0 서비스0방문목적가족외식(1)편의시설무료주차(1...

24 이 식당의 첫 평가를 작성해 주세요다녀온 맛집을 평가하면 다른 사용자에게 많은 도움...

25 연관맛집마블마블죽전네거리 6.8km와규, 소고기 가람한우식육식당경산 24.1km한우...

26 이 식당의 첫 평가를 작성해 주세요다녀온 맛집을 평가하면 다른 사용자에게 많은 도움...

27 연관맛집장승마을신탄진 22.1km게장, 간장게장 옥돌돼지갈비천안 31.9km돼지갈비...

맛집 리뷰 데이터 수집

2. 데이터 수집 및 전처리

데이터 전처리 >

2) 불용어 처리 3) mecab 사용자 사전 추가 4) 토큰화

```
with open('불용어리스트.txt', "r", encoding='UTF-8') as file:
    strings = file.readlines()
    #print(strings)

불용어=pd.DataFrame({'불용어':strings})
불용어=불용어['불용어'].str.replace('\n','')
stopwords=불용어.to_list()

import urllib.request
from konlpy.tag import Okt
import pandas as pd
import matplotlib.pyplot as plt
from ckonlpy.tag import Twitter
twitter = Twitter()

C:\Users\user\anaconda3\lib\site-packages\konlpy\tag\_okt.py:16: UserWarning: "Twitter" has changed to "Okt" since KoNLPy v0.4.5.
warn("'Twitter' has changed to 'Okt' since KoNLPy v0.4.5.")

twitter.add_dictionary(li, 'Noun')

tokenized_data = []
for sentence in re['리뷰']:
    temp_X = twitter.morphs(sentence, stem=True) # 토큰화
    temp_X = [word for word in temp_X if not word in stopwords] # 불용어 제거
    tokenized_data.append(temp_X)
    # print('불용어=>',temp_X)

len(tokenized_data)

20503
```

불용어 처리 및 토큰화

```
import pandas as pd
df=pd.read_csv('줄임말리스트.csv')
df
```

	줄임말
0	평냉
1	비냉
2	물냉
3	완냉
4	찍먹
...	...
94	레게노
95	레전드
96	재방문
97	캐주얼
98	서민적인

키워드 사전 추가

2. 데이터 수집 및 전처리

데이터 전처리 >

2) 불용어 처리 3) mecab 사용자 사전 추가 4) 토큰화

토큰화 : OKT

형태소 분석 : Mecab

```
with open('불용어리스트.txt', "r", encoding='UTF-8') as file:
    strings = file.readlines()
    #print(strings)

불용어=pd.DataFrame({'불용어':strings})
불용어=불용어['불용어'].str.replace('\n','')
stopwords=불용어.to_list()

import urllib.request
from konlpy.tag import Okt
import pandas as pd
import matplotlib.pyplot as plt
from ckonlpy.tag import Twitter
twitter = Twitter()

C:\Users\user\anaconda3\lib\site-packages\urllib\t\oknlp\16: warning: "Twitter" has changed to "Okt" since KoNLPI 0.4.0.

twitter.add_dictionary(li, 'Noun')

tokenized_data = []
for sentence in re['리뷰']:
    temp_X = twitter.morphs(sentence, stem=True) # 토큰화
    temp_X = [word for word in temp_X if not word in stopwords] # 불용어 제거
    tokenized_data.append(temp_X)
# print('불용어=>', temp_X)

len(tokenized_data)

20503
```

불용어 처리 및 토큰화

```
import pandas as pd
df=pd.read_csv('줄임말리스트.csv')
df
```

	줄임말
0	평냉
1	비냉
2	물냉
3	완냉
94	레게노
95	레전드
96	재방문
97	캐주얼
98	서민적인

키워드 사전 추가

2. 데이터 수집 및 전처리

데이터 전처리 > 토큰화 라이브러리 비교

```
import urllib.request
from konlpy.tag import Okt
import pandas as pd
okt=Okt()
okt.morphs(sent)

['내',
'가',
'먹어',
'본',
'냉면',
'중',
'가장',
'맛있는',
'냉면',
'이었다',
'.',
'면발',
'이',
'쫄깃쫄깃',
'하고',
'탱글탱글했고',
'안',
'그래도',
'더웠는데',
'시원하니',
'잘',
'먹었다']
```

Okt

```
from soynlp.tokenizer import MaxScoreTokenizer

maxscore_tokenizer = MaxScoreTokenizer(scores=scores)
maxscore_tokenizer.tokenize(sent)

['내가',
'먹어',
'본',
'냉면중',
'가장',
'맛있는',
'냉면이었다.',
'면발이',
'쫄깃',
'쫄깃',
'하고',
'탱글탱글했고',
'안',
'그래도',
'더웠는데',
'시원하니',
'잘',
'먹었다',
'.굿굿']
```

soynlp

```
from eunjeon import Mecab
m=Mecab()
m.morphs(sent)

['내',
'가',
'먹',
'어',
'본',
'냉면',
'중',
'가장',
'맛있',
'는',
'냉면',
'이',
'었',
'다',
'.',
'면발',
'이',
'쫄깃쫄깃',
'하',
'고',
'탱글탱글',
'했',
'고',
'안',
'그래도',
'더웠',
'는데',
'시원',
'하',
'니',
'잘',
'먹',
'었',
'다']
```

mecab

2. 데이터 수집 및 전처리

데이터 전처리 >

5) 정규표현식 특수문자 제거 6) 맛집 키워드 추출

1	df['리뷰']=df['리뷰'].str.replace("[^ㄱ-ㅎㅏ-ㅣ가-힣]", "")
2	df['리뷰']=df['리뷰'].str.replace(' ', '')
3	df['리뷰']=df['리뷰'].str.replace('ㅋ', '')
4	df['리뷰']=df['리뷰'].str.replace('ㅎ', '')
5	df['리뷰']=df['리뷰'].str.replace('ㅌ', '')
6	df['리뷰']=df['리뷰'].str.replace('ㅠ', '')
7	df['리뷰']

1	review_clear = []
2	for i in range(len(df3['리뷰'])):
3	ab = df3['리뷰'][i].replace("연관맛집", "")
4	review_clear.append(ab)
5	abb = df3['리뷰'][i].replace("블로그리뷰", "")
6	review_clear.append(abb)
7	abbb = re.sub('[^가-힣]', ' ', df3['리뷰'][i])
8	review_clear.append(ab)

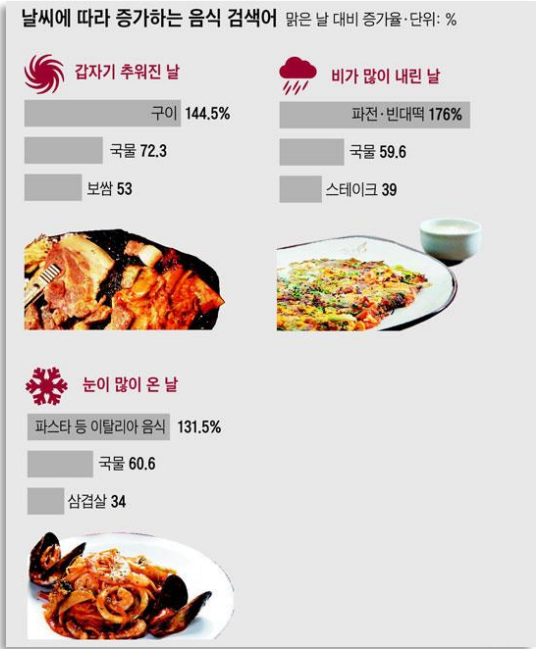
정규표현식 특수문자 제거

상호ID	리뷰	목적	먹집 키워드	대표메뉴
0 S0000002	점심식사분위기캐주얼한매우만족만족보통만매우불만평가 참여율이 부족합니다 평가에 참여해...	[점심식사]	[두통, 맛집, 캐주얼]	[연어]
1 S0000003	식사모임 기념일 실버푸드분위기서민적인 경관아경이 좋은 숨은맛집 편의시설무료주차 개별룸...	[집대, 가족, 기념일, 점심식사, 모임]	[맛있다, 맛집, 좋다, 가성비, 서민적인]	[소갈비, 돼지갈비, 갈비탕]
2 S0000004	사당나인피스그릴 삼겹살에 버터 조합이라니언제나 먹어도 맛있는 삼겹살 사당역 맛집을...	[점심, 모임]	[고급, 추천, 맛있다, 맛집, 빨갳다, 좋다, 굽다, 유명, 존맛탱, 존맛]	[삼겹살, 스테이크, 돼지고기]
3 S0000005	회식 술모임 집대 가족외식분위기지역주민이 찾는 숨은맛집 가성비좋은편의시설칸막이 무료...	[집대, 회식, 가족, 혼밥, 모임]	[물냉, 짬뽕하다, 달달, 적다, 맛있다, 맛집, 좋다, 비냉, 가성비, 서민적인]	[냉면, 등심, 돼지갈비]
4 S0000008	저녁식사 술모임분위기가성비좋은 서민적인매우만족만족보통만매우불만평가 참여율이 부족함...	[모임]	[괜찮다, 맛있다, 맛집, 좋다, 가성비, 서민적인]	[갈비, 돼지갈비]
...
20498 S0034701	송파구 맛집 여름엔 냉면이지 함흥평양냉면 전문점 당고냉면너무 덥다 정말 너무 더...	[점심]	[시원, 평범, 달콤, 짜다, 맛집]	[갈비찜, 냉면]
20499 S0034702	방이샤브샤브칼국수대치동 칼국수 진성칼국수청담동 칼국수 보쌈정식 어바웃샤브샤브...	[]	[]	[샤브샤브, 보쌈, 칼국수]
20500 S0034703	저녁식사 실버푸드 데이트분위기지역주민이 찾는 푸짐한 숨은맛집편의시설발렛주차매우만족만...	[점심식사, 데이트]	[비싸다, 시골벽적, 맛있다, 맛집, 쏘쏘, 나쁘다, 서민적인]	[파전]
20501 S0034704	점심식사분위기고급스러운 예쁜 푸짐한편의시설무료주차매우만족만족보통불만매우불만평가참여율...	[점심식사]	[맛집, 많이, 예쁘다, 고급]	[]
20502 S0034706	점심식사 술모임 저녁식사분위기시골벽적인 캐주얼한 숨은맛집편의시설야외좌석테라스매우만족...	[가족, 회식, 점심식사, 모임]	[캐주얼, 시골벽적, 맛있다, 테라스, 맛집, 꿀맛, 좋다, 부드럽다, 매콤, 재방...	[오랭]

리뷰에서 맛집 키워드 추출

2. 데이터 수집 및 전처리

추천 시스템 자료 수집



대분류	소분류	총인원 (명)	날씨영향 받은	백분율(%)	날씨영향 받지않음	백분율(%)
성별	남	89	58	65.2	31	34.8
	여	265	187	70.6	78	29.4
연령별	1~19세	63	36	57.1	27	42.9
	20~29세	188	137	72.9	51	27.1
	30~39세	72	51	70.8	21	29.2
	40~49세	23	15	65.2	8	34.8
	50세 이상	8	6	75	2	25
건강상태	건강	272	187	68.8	85	31.2
	허약	82	58	70.7	24	29.3
몸무게	정상	227	164	72.2	63	27.8
	비만	90	55	61.1	35	38.9
	저체중	37	26	70.3	11	29.7

특정 음식이 땡기는 이유

단 맛 ----- 에너지 부족

매운 맛 ---- 스트레스 과다

유제품 ----- 심리적 불안

짭 맛 ----- 영양 부족

신 맛 ----- 긴장 이완 필요

커피, 탄산 ----- 피로 누적

과일 ----- 기분전환이 필요

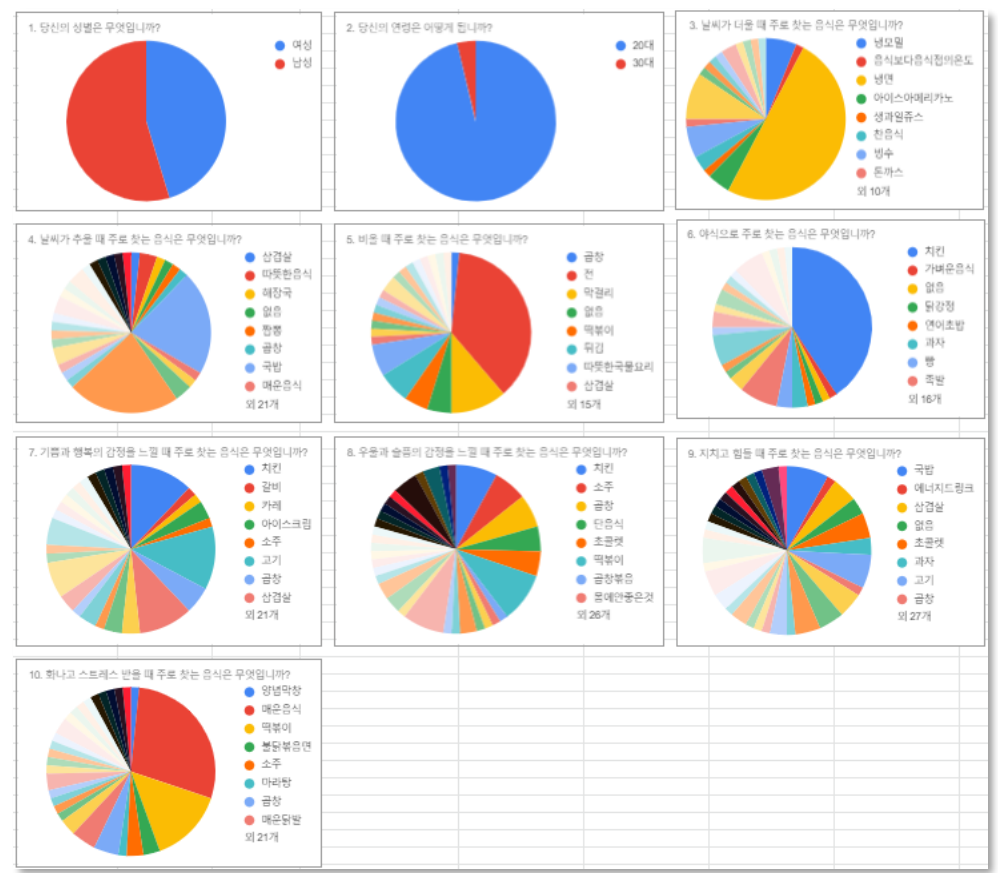
*참고문헌 : 스마트 기반 날씨정보 콘텐츠 개발 연구:날씨에 따른 위치기반 음식 추천 애플리케이션 개발

2. 데이터 수집 및 전처리

추천 시스템 자료 수집

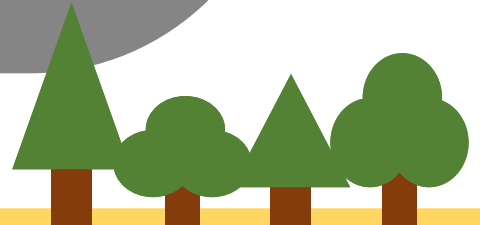
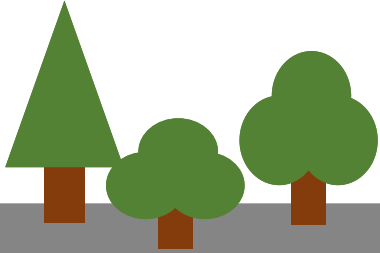
타임스탬프	1. 당신의 성별은 무엇입니까?	2. 당신의 연령은 어떻게 될3. 날씨가 더울 때 주로 찾4. 날씨가 추울 때 주로 찾5. 비올 때 주로 찾는 음식6. 야식으로 주로 찾는 음식7. 기쁨과 행복의 감정을 느8. 우울과 슬픔의						
9-16-2020 15:19:51	여성	20대	냉모밀	삼겹살	곱창	치킨	치킨	치킨
9-16-2020 15:22:29	남성	20대	음식보다 음식점의 온도	몸을 녹일 수 있는 뜨끈한	파전에 막걸리	배가너무 부르지 않는 가벼	갈비	소주
9-16-2020 15:23:09	남성	20대	냉면	빠해장국	파전	치킨	카레	곱창
9-16-2020 15:24:18	남성	30대	냉면	그런거 없음	없음	안먹음	아이스크림	단것
9-16-2020 15:27:30	여성	20대	아이스아메리카노, 생과일	짬뽕, 곱창구이	떡볶이, 튀김	닭강정, 연어초밥	소주, 고기	소주
9-16-2020 15:28:13	남성	20대	찬 음식	따뜻한 음식	따뜻한 국물	과자 & 빵	곱창	초콜렛
9-16-2020 15:29:27	남성	20대	냉면	수육국밥	전	치킨	삼겹살	소주
9-16-2020 15:29:51	남성	20대	시원한음식	매운음식	전	족발, 치킨	없음	달달한 음식
9-16-2020 15:30:43	여성	20대	냉면	나베	전	치킨	고기	치킨
9-16-2020 15:31:59	남성	20대	팔뚝수	라면	전	치킨	치킨	치킨, 떡볶이
9-16-2020 15:32:23	여성	20대	빙수	국물요리	국물요리, 전, 삼겹살	떡볶이, 닭발, 치킨	떡볶이	엽기떡볶이
9-16-2020 15:32:32	여성	20대	냉모밀&돈까스	오리백숙	떡볶이	치킨	닭강정	곱창볶음
9-16-2020 15:40:47	남성	20대	아이스크림 냉면 몸보신	뜨끈한 국물	전	치킨	===건강식	몸에 안좋은 것
9-16-2020 15:42:53	여성	20대	아이스크림이랑 아이스아	뜨거운 아메리카노	김치전 !	치킨	기쁘고 행복할때는 음식이	슈크림 / 맵고 짜
9-16-2020 15:48:37	남성	20대	냉면	튀김	오징어회	족발	김치볶음밥	소주
9-16-2020 15:48:51	여성	20대	빙수, 냉면	탕류	막걸리	집에 있는거 아무거나	고기류	과자
9-16-2020 15:51:12	여성	20대	냉면	짜개류	전	치킨	치킨	짜장면
9-16-2020 15:55:06	남성	20대	냉면, 아이스크림	칼국수	파전에 막걸리, 전 종류	치킨, 피자, 맥주	피자	떡볶이 같은 매운
9-16-2020 15:55:32	여성	20대	비빔국수	따뜻한 국물류(전골, 칼국	파전 찜꾸미	마라탕	케이크	매운음식 혹은 단
9-16-2020 15:56:28	남성	20대	냉면	국물요리류	배달음식	치킨	삼겹살	두루치기
9-16-2020 16:04:33	여성	20대	떡볶이	쌀국수	쌀국수	치킨	떡볶이	떡볶이
9-16-2020 16:04:36	여성	20대	냉면	국물있는음식	칼국수	치킨 족발	양식 브런치 파스타	대충 먹음
9-16-2020 16:05:01	남성	20대	냉면	국밥	짜장면	치킨, 피자	케이크	패스트푸드
9-16-2020 16:05:31	남성	20대	냉면	국밥	국밥	햄버거	아이스크림	아이스크림
9-16-2020 16:07:30	남성	20대	냉면	떡볶이	파전	곱창	파스타	편의점
9-16-2020 16:10:22	남성	20대	냉면	국밥	국밥	치킨	국밥	국밥
9-16-2020 16:16:27	남성	20대	아이스크림	편지	파전	나눔	빵	초콜렛

2020.9.16 설문조사 결과 자료



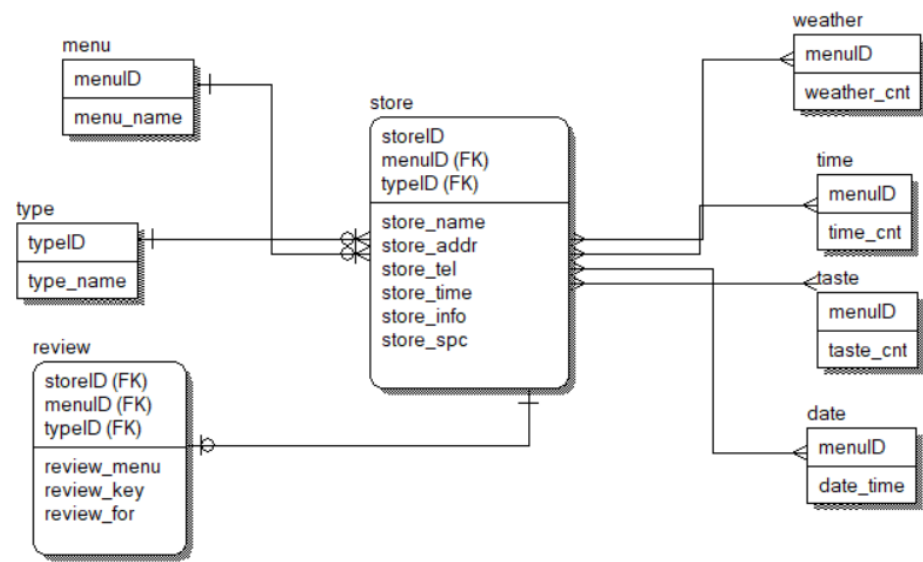
2020.9.16 설문조사 결과 정리 자료

3. 개발 프로세스



3. 개발 프로세스

DB 설계 > RDB



Erwin을 사용한 맛집 DB 설계

Table: store

Columns:

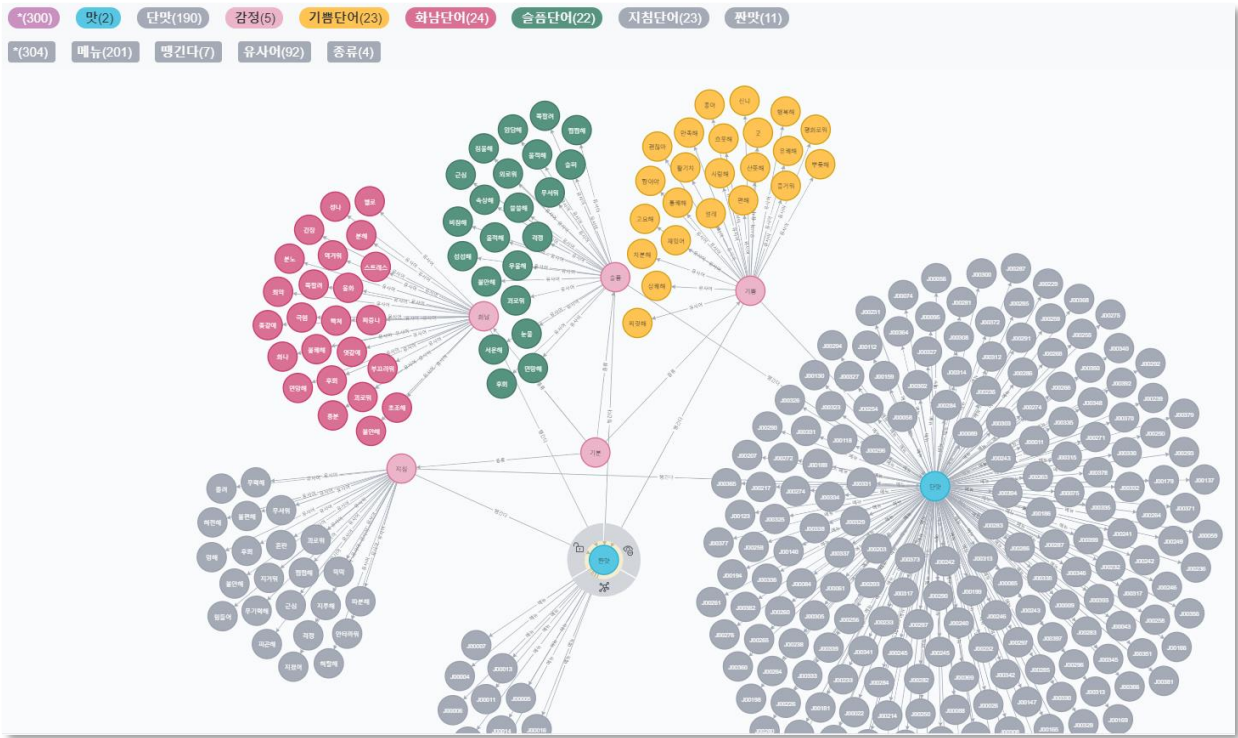
- storeID: PK, varchar(45)
- menuID: varchar(1000)
- store_name: varchar(1000)
- store_addr: varchar(1000)
- store_tel: varchar(1000)
- store_lon: varchar(1000)
- store_menu: varchar(45)
- store_info: varchar(1000)
- store_spc: varchar(1000)
- store_addr1: varchar(1000)
- store_addr2: varchar(1000)
- store_addr3: varchar(1000)
- review_keyword: varchar(1000)
- review_for: varchar(1000)
- review_menu: varchar(1000)
- store_tel: varchar(1000)

Result Grid:

storeID	menuID	store_name	store_addr	store_tel	store_lon	store_menu
S0000003	300250	526	서울 노원구 공릉로 112지번공릉동 644-3	526	37.6202	127.0781
S0000004	300201	853	서울특별시 중랑구 인사동12길 16지번서울시...		37.5747	126.9851
S0000005	300250	1842	서울특별시 용산구 이태원로 244 트윈빌딩지번...		37.5365	127.0005
S0000006	300251	3140	서울특별시 송파구 방이동 62-7		37.5136	127.1093
S0000007	300377	5412	서울 용산구 이태원로54가길 12지번한남동 683...		37.5373	127.0017
S0000008	300246	72420	서울특별시 송파구 백제고분로45길 12지번서...		37.5096	127.1102
S0000009	300361	8meal	서울 성동구 왕십리로4길 26-18지번성수동1가...		37.5468	127.0469
S0000010	300232	03수제비합국수	서울특별시 서초구 방배동 768-2		37.4951	126.9854
S0000011	300279	03수제비합국수	서울특별시 서초구 방배동 768-2		37.4951	126.9854
S0000019	300182	100년설렁탕	서울특별시 성북구 삼선동2가 38		37.5881	127.0099
S0000020	300215	100년의곰 제...	서울특별시 강북구 돈암동 603-1		37.5357	127.1472
S0000023	300312	100쪽발	서울 동작구 사당로30길 154지번사당동 1041-2...		37.478	126.9806
S0000024	300087	101번지 남산...	서울특별시 중구 남산동2가 49-24		37.557	126.9857
S0000029	300338	10고르소고모	서울특별시 강남구 압구정로 416지번서울시 강...		37.5269	127.0424
S0000033	300379	117 파크	서울특별시 중랑구 경희궁1길 16지번서울시 중...		37.5717	126.9707
S0000034	300182	117파크	서울특별시 중랑구 신문로2가 1-117		37.5717	126.9707
S0000035	300220	117파크	서울특별시 중랑구 신문로2가 1-117		37.5717	126.9707
S0000036	300070	119닭발	서울특별시 강동구 천호2동 307-9		37.545	127.1302

Mysql DB 구현

DB 설계 > Graph DB



Neo4j를 사용한 기분 별 메뉴 DB

```
from neo4j import GraphDatabase
```

```
def run_query(func):
    uri = 'neo4j://localhost:7687'
    user = 'neo4j'
    password = '1234'
    driver = GraphDatabase.driver(uri, auth=(user, password))
    with driver.session() as session:
        session.read_transaction(func)
```

```
lst = []
def get_feelings(tx):
    query = """
        MATCH (a:감정{기분:'기분'}) - [r:종류] -> (b)
        RETURN b.기분 AS 기분의종류
    """
    result = tx.run(query)

    for record in result:
        lst.append(record['기분의종류'])

    print('기분의종류에는', ', '.join(lst), '이 있습니다.')

run_query(get_feelings)
```

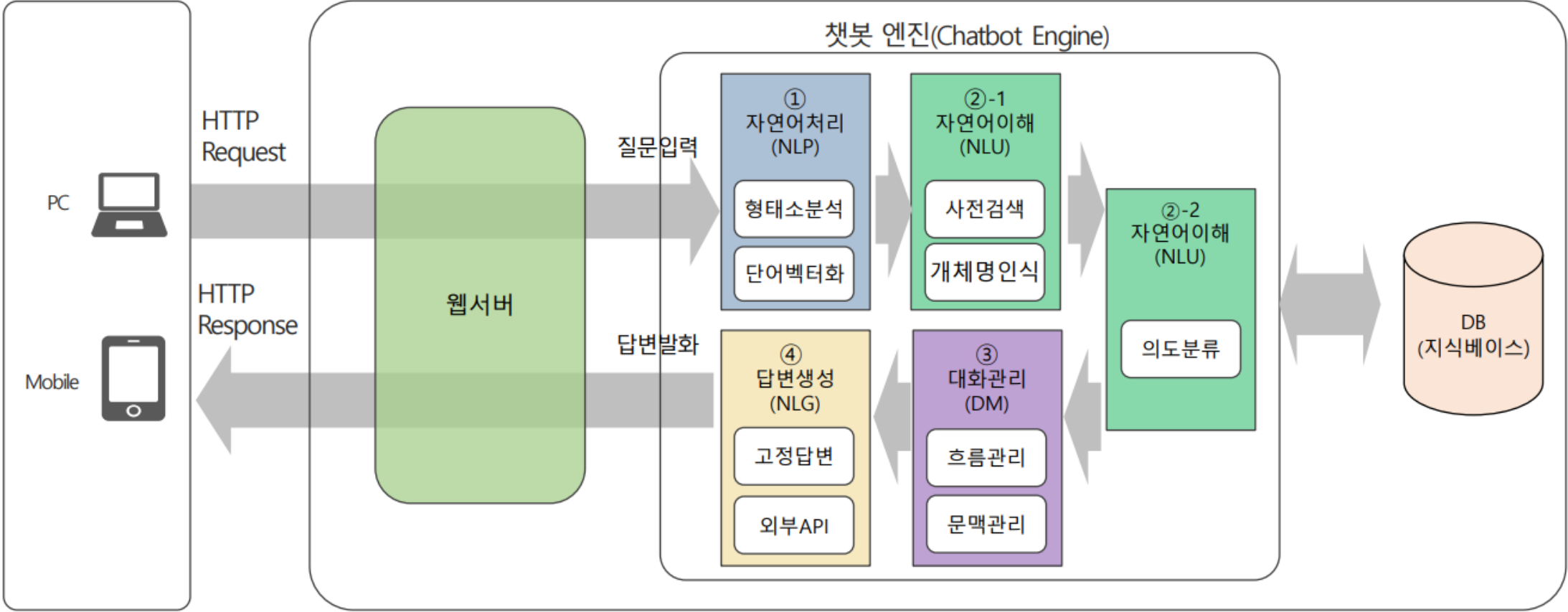
기분의종류에는 지침, 슬픔, 화남, 기쁨 이 있습니다.

DB 설계 > G

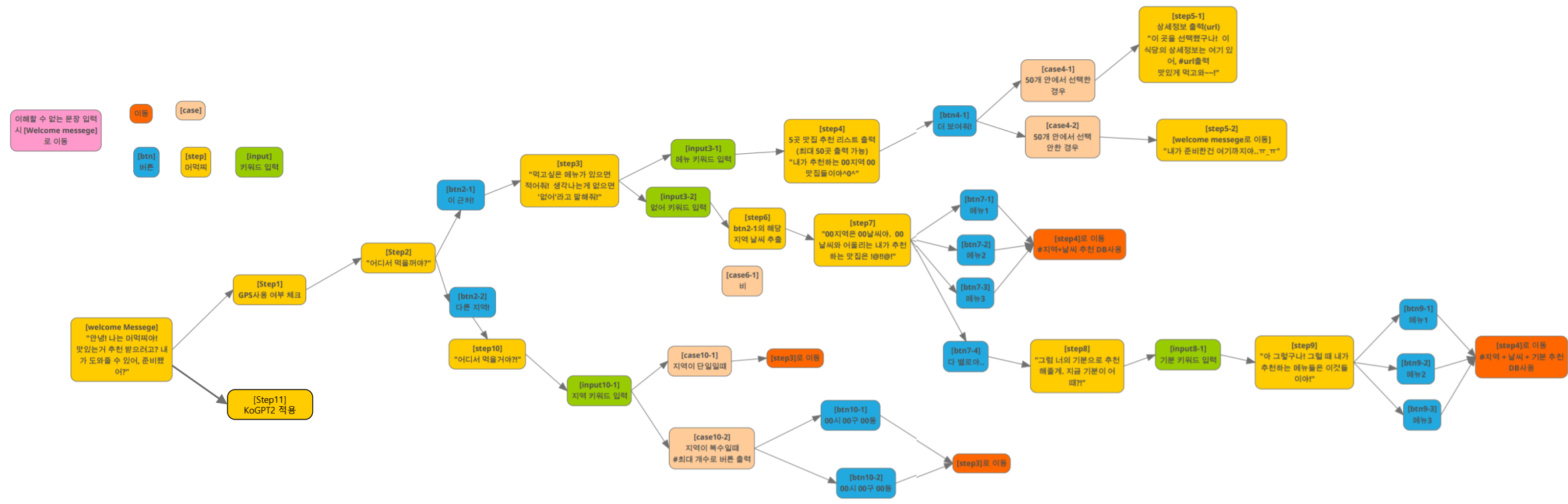
챗봇 구성도

챗봇 클라이언트(Chatbot Client)

챗봇 서버 (Chatbot Server)



챗봇 시나리오 설계



MindMup2.0을 활용한 챗봇 플로우 설계

챗봇

프론트 엔드

: 텔레그램 챗봇 API 활용

챗봇 및 추천 엔진

: 총 9개의 핵심 Task로 구성

(딥러닝 : 3, API : 2, Rule base 및 기타 : 4)

3. 개발 프로세스

기술 적용

구분	명칭	수행 내용
DL	감정분석(SA)	사용자 리뷰의 긍/부정 분류
	개체명 인식	LSTM + CRF 활용
	인공지능	GPT2

상세
Bert 모델 이용한 네이버 영화리뷰 감정 분석
한국해양대학교 자연어처리 연구실 제공 ner_kor_dataset 활용
AWS + SKT에서 제공한koGPT2를 이용한 대화 처리

```
In [13]: model = Sequential()
model.add(Embedding(input_dim=vocab_size, output_dim=20, input_length=max_len, mask_zero=True))
model.add(Bidirectional(LSTM(units=50, return_sequences=True, recurrent_dropout=0.1)))
model.add(TimeDistributed(Dense(50, activation="relu"))) #각 스텝마다 cost (오류)를 계산해서 하위 스텝으로 전달
crf = CRF(tag_size)
model.add(crf)

In [20]: model.compile(optimizer="adam", loss=crf.loss,
, metrics=[crf.accuracy])
history = model.fit(X_train, y_train, batch_size=32, epochs=10, validation_split=0.01, verbose=1)

Epoch 1/10
573/573 [=====] - 106s 185ms/step - loss: 0.3757 - accuracy: 0.9700
- val_loss: 17.8494 - val_accuracy: 0.9044
Epoch 5/10
573/573 [=====] - 103s 180ms/step - loss: 0.3606 - accuracy: 0.9711
- val_loss: 18.0458 - val_accuracy: 0.9044
Epoch 6/10
573/573 [=====] - 103s 180ms/step - loss: 0.3466 - accuracy: 0.9721
- val_loss: 18.1298 - val_accuracy: 0.9061
Epoch 7/10
573/573 [=====] - 103s 180ms/step - loss: 0.3343 - accuracy: 0.9728
- val_loss: 18.3590 - val_accuracy: 0.9030
Epoch 8/10
573/573 [=====] - 103s 181ms/step - loss: 0.3196 - accuracy: 0.9740
- val_loss: 18.5485 - val_accuracy: 0.9023
Epoch 9/10
573/573 [=====] - 105s 184ms/step - loss: 0.3051 - accuracy: 0.9781
- val_loss: 18.7846 - val_accuracy: 0.8988
Epoch 10/10
573/573 [=====] - 103s 180ms/step - loss: 0.2913 - accuracy: 0.9793

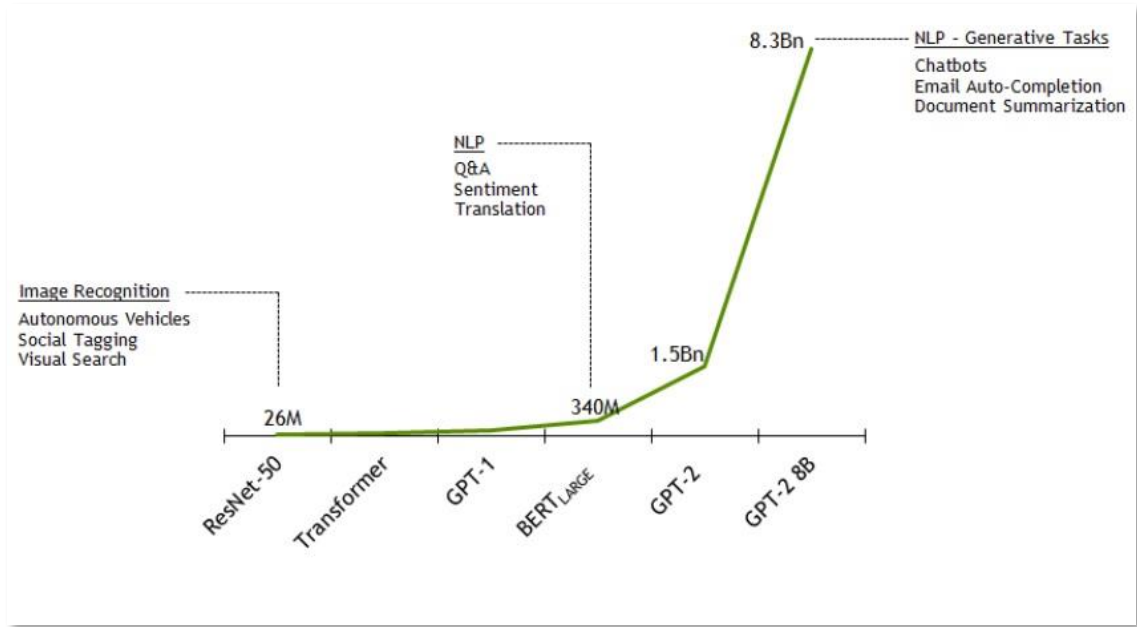
In [21]: print("\n 테스트 정확도: %.4f" % (model.evaluate(X_test, y_test)[1]))

145/145 [=====] - 10s 66ms/step - loss: 18.0752 - accuracy: 0.8927

테스트 정확도: 0.8927
```

LSTM + CRF 모델을 활용한 개체명 인식

GPT2 (Generative Pre-Training Model2)



```
# 사전훈련된 koGPT2를 챗봇 데이터로 파인튜닝
CUDA_VISIBLE_DEVICES=0 python train.py --num-epoch 2 --train

D-
[REDACTED]

using cached model
INFO:root:contexts : 침입하다
INFO:root:token ctb: ['<usr>', '침입', '하다', '</s>', '<unused>', '_1', '</s>']
INFO:root:response : 기분이 많이 안 좋겠어요
INFO:root:token response: ['<sys>', '_기분이나', '_안', '_좋', '겠어요', '.', '</s>']
INFO:root:labels ['<unused>', '<unused>', '<unused>', '<unused>', '<unused>', '<unused>', '<unused>', '_기분이나', '_안', '_좋', '겠어요', '.', '</s>']
INFO:root:contexts : 헤어지지 3주차 넘어가네-
INFO:root:token ctb: ['<usr>', '_헤어진', '지', '_3', '주차', '_넘어가', '네', '!', '</s>', '<unused>', '_1', '</s>']
INFO:root:response : 잘 견디시길 바라요
INFO:root:token response: ['<sys>', '_잘', '_견디', '네', '길', '_바라', '요', '!', '</s>']
INFO:root:labels ['<unused>', '<unused>', '<unused>', '<unused>', '<unused>', '<unused>', '<unused>', '<unused>', '<unused>', '<unused>', '<unused>', '_잘', '_견디', '네', '길', '_바라', '요', '!', '</s>']
INFO:root:contexts : 연금 믿어도 될까
INFO:root:token ctb: ['<usr>', '_연금에', '_일확', '_일어나', '서', '_연금', '_해하', '겠', '어', '!', '</s>', '<unused>', '_0', '</s>']
INFO:root:response : 일확 일어나는 새가 볼래들 더 많이 먹죠
INFO:root:token response: ['<sys>', '_일확', '_일어나는', '_새', '가', '_볼', '_래들', '_더', '_많이', '_먹', '죠', '!', '</s>']
INFO:root:labels ['<unused>', '<unused>', '<unused>', '<unused>', '<unused>', '<unused>', '<unused>', '<unused>', '<unused>', '<unused>', '<unused>', '_일확', '_일어나는', '_새', '_먹', '_일어나는', '!', '</s>']
INFO:root:token ctb: ['<usr>', '_연금', '_믿', '어도', '_될', '까', '</s>', '<unused>', '_0', '</s>']
INFO:root:response : 없는 것보다 나은 거예요
INFO:root:token response: ['<sys>', '_없는', '_것보다', '_나', '을', '_거예요', '!', '</s>']
INFO:root:labels ['<unused>', '<unused>', '<unused>', '<unused>', '<unused>', '<unused>', '<unused>', '<unused>', '<unused>', '<unused>', '<unused>', '_없는', '_것보다', '_나', '을', '_거예요', '!', '</s>']
INFO:root:contexts : 개강웃 예쁘게 입어 볼까
INFO:root:token ctb: ['<usr>', '_개강', '웃', '_예쁘', '게', '_입어', '볼', '까', '</s>', '<unused>', '_0', '</s>']
INFO:root:response : 개시해보세요
INFO:root:token response: ['<sys>', '_개시', '해', '_보세요', '!', '</s>']
INFO:root:labels ['<unused>', '<unused>', '<unused>', '<unused>', '<unused>', '<unused>', '<unused>', '<unused>', '<unused>', '<unused>', '<unused>', '_개시', '해', '_보세요', '!', '</s>']
[Epoch 1 Batch 50/185] loss=1.0511, lr=0.0000480480, train ppl=2.861
[Epoch 1 Batch 100/185] loss=0.8741, lr=0.0000405405, train ppl=2.649
[Epoch 1 Batch 150/185] loss=0.8544, lr=0.0000393030, train ppl=2.597
[Epoch 2 Batch 15/185] loss=0.8894, lr=0.000035255, train ppl=1.328
[Epoch 2 Batch 65/185] loss=0.8558, lr=0.0000180180, train ppl=2.601
[Epoch 2 Batch 115/185] loss=0.8822, lr=0.0000105105, train ppl=2.618
[Epoch 2 Batch 165/185] loss=0.8517, lr=0.0000000000, train ppl=2.590
INFO:root:saving model file to koGPT2_chat.params
```

GPT-2는 15억 개 파라미터를 가진 대규모 번역기반 언어 모델로,
800만개 텍스트의 데이터셋을 통해 학습하여 임의의 문장을 입력하면
맥락을 고려한 종합적인 문장을 스스로 생성

AWS와 SKT에서 제공한 koGPT2 Train 코드

3. 개발 프로세스

기술 적용

구분	명칭	수행 내용
API	날씨	공공데이터포털 동네예보 조회서비스 API 활용
	위경도	카카오 오픈 빌더 활용

```
import requests
import json
import datetime

vilage_weather_url = "http://apis.data.go.kr/1360000/VilageFcstInfoService/getVilageFcst?"

service_key =

today = datetime.datetime.today()
base_date = today.strftime("%Y%m%d") # "20200214" == 기존 날짜
base_time = "0800" # 날씨 값

# 마포구 위도, 경도
nx = '59'
ny = '127'
# nx = a
# ny = b

payload = "serviceKey=" + service_key + "&" + \
    "dataType=json" + "&" + \
    "base_date=" + base_date + "&" + \
    "base_time=" + base_time + "&" + \
    "nx=" + nx + "&" + \
    "ny=" + ny

# GET 요청
res = requests.get(vilage_weather_url + payload)
item=res.json().get('response').get('body').get('items')
item

...

# sky 상태 : 맑음(0), 비(1), 비/눈(눈개비)(2), 눈(3), 소나기(4), 빗방울(5), 빗방울/눈날림(6), 눈날림(7)
sky=item['item'][1]['fcstValue']
sky

'0'

# temp 상태 : 현재 3시간내의 기온
temp=item['item'][6]['fcstValue']
int(temp)

13
```

날씨 추출

```
def address_to_latlon(address):
    url = "https://dapi.kakao.com/v2/local/search/address.json?query=" + address
    result = requests.get(urlparse(url).geturl(),
        headers={"Authorization":"KakaoAK cec1e1df2db071b9babb279c05b80672"})#API 날기
    json_obj = result.json()
    for document in json_obj['documents']:
        global a
        a = [document['address_name'], document['y'], document['x']]
    return a

address_to_latlon("서울 마포구 양화로183")

['서울 마포구 양화로 183', '37.5583675467575', '126.925148412411']
```

주소 → 위경도 추출

3. 개발 프로세스

기술 적용

구분	명칭	수행 내용
Rule base 및 기타	대화 관리(DM)	규칙 기반 + DL 활용 대화 흐름 관리
	고정 답변 생성	고정 답변을 생성
	분석 / 재학습 모듈	사용자 대화 수집과 학습
	레벤슈타인 유사도 측정	사용자 입력 오타 교정

```
from soynlp.hangle import levenshtein
from soynlp.hangle import jamo_levenshtein
def levenshtein(word):
    s1 = word
    diction={}
    for i in word_list:
        lv=jamo_levenshtein(s1, i)
        if lv <= 1:
            diction[i] = lv
            #print(diction)
            # print(i, ':', lv)
    # print(len(diction))
    if len(diction) > 0:
        key_min=min(diction.keys(),key=(lambda k: diction[k]))
    else:
        key_min=None
    return key_min

love=levenshtein('삼겹살ㄹ')
love

'삼겹살'
```

레벤슈타인 유사도 분석

3. 개발 프로세스

기술 적용

구분	명칭	수행 내용	상세
DL	감정분석(SA)	사용자 리뷰의 긍/부정 분류	네이버 영화리뷰 긍/부정 데이터 셋
	개체명 인식	LSTM + CRF 활용	한국해양대학교 자연어처리 연구실 제공 ner_kor_dataset
	인공지능	koGPT2를 이용한 대화 처리	
API	날씨	공공데이터포털 동네예보 조회서비스 API 활용	
	위경도	카카오 오픈 빌더 활용	
Rule base 및 기타	대화관리(DM)	규칙 기반 + DL 활용 대화 흐름 관리	
	고정 답변 생성	고정 답변을 생성	
	분석 / 재학습 모듈	사용자 대화 수집과 학습	
	레벤슈타인 유사도 측정	사용자 입력 오타 교정	

4. 시연



4. 시연

def handler1(bot, update):
 text = update.message.text
 chat_id = update.message.chat_id
 print('gpt')
 mm = get_menu(text)
 if callquery[0] =
 if mm is not
 menus.append(
 bot.send_

 print(men
 df1_gu =
 df2 = get

 df2 = df2
 print(df2)
 # df2를 랜

 userlist =
 if len(df
 bot.s
 # Str

 menus =
 addrs =
 gu_li
 bot.s

 else:
 df3 = tuple(divide_list(df2, 5))
 print(df3)
 print(len(df3))
 # 1번(df3)을 구

← M meomeokzzi
봇

이 봇은 무엇을 할 수 있나요?
"안녕, 나는 머먹찌! 맛있는 거 어디서
먹을거야? >ㅅ<"

11월 5일

난 맛집 추천봇 머먹찌야! 🐹
나랑 뭐할래?
처음으로 돌아가고 싶다면
머먹찌 라고 쳐줘! 오후 1:44

맛집 추천받기

머먹찌랑 대화하기

메시지

장소 > 메뉴

def handler1(bot, update):
 text = update.message.text
 chat_id = update.message.chat_id
 print('text')
 callquery = []
 def handler1(bot, update):
 text = update.message.text
 chat_id = update.message.chat_id
 print('text')
 '맛집 추천받기':
 me_:
 id(text)
 message(chat_id=chat_id,
 text=text + ' 먹고싶구나?! 추천해줄게!' + '\n'
)
 et_tablebylocation(addr[0])
 tablebymenu(menus[0], df1)
 values.tolist()
 으로 5개씩 묶어 놓는 함수 만들기 OK
 [2, 3, 6, 7, 8, 9, 10, 14, 15, 16, 17]
 == 0:
 id_message(chat_id=chat_id, text='미안.. 아직 내 디
 essage.text = "" + update.getMessage().get
 clear()
 clear()
 clear()
 idMessage(chat_id=chat_id,
 text=emojize(
 "다시 안녕! 나는 머먹찌야:hamster:" +
 use_aliases=True))
 로 돌아가기 구현. OK
 else:
 df3 = tuple(divide_list(df2, 5))
 print(df3)
 print(len(df3))
 # 1번(df3)을 구

← M meomeokzzi
봇

이 봇은 무엇을 할 수 있나요?
"안녕, 나는 머먹찌! 맛있는 거 어디서
먹을거야? >ㅅ<"

11월 5일

난 맛집 추천봇 머먹찌야! 🐹
나랑 뭐할래?
처음으로 돌아가고 싶다면
머먹찌 라고 쳐줘! 오후 2:28

맛집 추천받기

머먹찌랑 대화하기

메시지

장소 > 날씨 > 기본

def handler1(bot, update):
 text = update.message.text
 chat_id = update.message.chat_id
 print('text')
 callquery = []
 def handler1(bot, update):
 text = update.message.text
 chat_id = update.message.chat_id
 print('text')
 '맛집 추천받기':
 me_:
 id(text)
 message(chat_id=chat_id,
 text=text + ' 먹고싶구나?! 추천해줄게!' + '\n'
)
 et_tablebylocation(addr[0])
 tablebymenu(menus[0], df1)
 values.tolist()
 으로 5개씩 묶어 놓는 함수 만들기 OK
 [2, 3, 6, 7, 8, 9, 10, 14, 15, 16, 17]
 == 0:
 id_message(chat_id=chat_id, text='미안.. 아직 내 디
 essage.text = "" + update.getMessage().get
 clear()
 clear()
 clear()
 idMessage(chat_id=chat_id,
 text=emojize(
 "다시 안녕! 나는 머먹찌야:hamster:" +
 use_aliases=True))
 로 돌아가기 구현. OK
 else:
 df3 = tuple(divide_list(df2, 5))
 print(df3)
 print(len(df3))
 # 1번(df3)을 구

← M meomeokzzi
봇

이 봇은 무엇을 할 수 있나요?
"안녕, 나는 머먹찌! 맛있는 거 어디서
먹을거야? >ㅅ<"

11월 5일

난 맛집 추천봇 머먹찌야! 🐹
나랑 뭐할래?
처음으로 돌아가고 싶다면
머먹찌 라고 쳐줘! 오후 2:40

맛집 추천받기

머먹찌랑 대화하기

메시지

장소 > 날씨

4. 시연

defspypy x chatbot9.py x

161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199

맛집 추천

def hanuena(bot, update):
 text = update.message.text
 chat_id = update.message.chat_id
 print('gpt')
 mm = get_menu(text)
 if callquery[0] =
 if mm is not
 menus.append(
 bot.send_

 가져오고 있어~' + '\n' + '조금만 기다려줘~')

 print(men
 df1_gu =
 df2 = get

 df2 = df2
 print(df2
 # df2를 랜

 userlist =
 if len(df
 bot.s
 # Str

 menus.
 addrs
 gu_li
 bot.s

 난 맛집 추천봇 머먹찌야! 🐹
 나랑 뭐할래?
 처음으로 돌아가고 싶다면
 머먹찌 라고 쳐줘! 오후 1:44

 맛집 추천받기
 머먹찌랑 대화하기

 # 처음

장소 > 메뉴

← M meomeokzzi 봇

이 봇은 무엇을 할 수 있나요?
"안녕, 나는 머먹찌! 맛있는 거 어디서 먹을거야? >ㅅ<"

11월 5일

난 맛집 추천봇 머먹찌야! 🐹
나랑 뭐할래?
처음으로 돌아가고 싶다면
머먹찌 라고 쳐줘! 오후 2:28

맛집 추천받기
머먹찌랑 대화하기

메시지

장소 > 날씨 > 기본

← M meomeokzzi 봇

이 봇은 무엇을 할 수 있나요?
"안녕, 나는 머먹찌! 맛있는 거 어디서 먹을거야? >ㅅ<"

11월 5일

난 맛집 추천봇 머먹찌야! 🐹
나랑 뭐할래?
처음으로 돌아가고 싶다면
머먹찌 라고 쳐줘! 오후 2:40

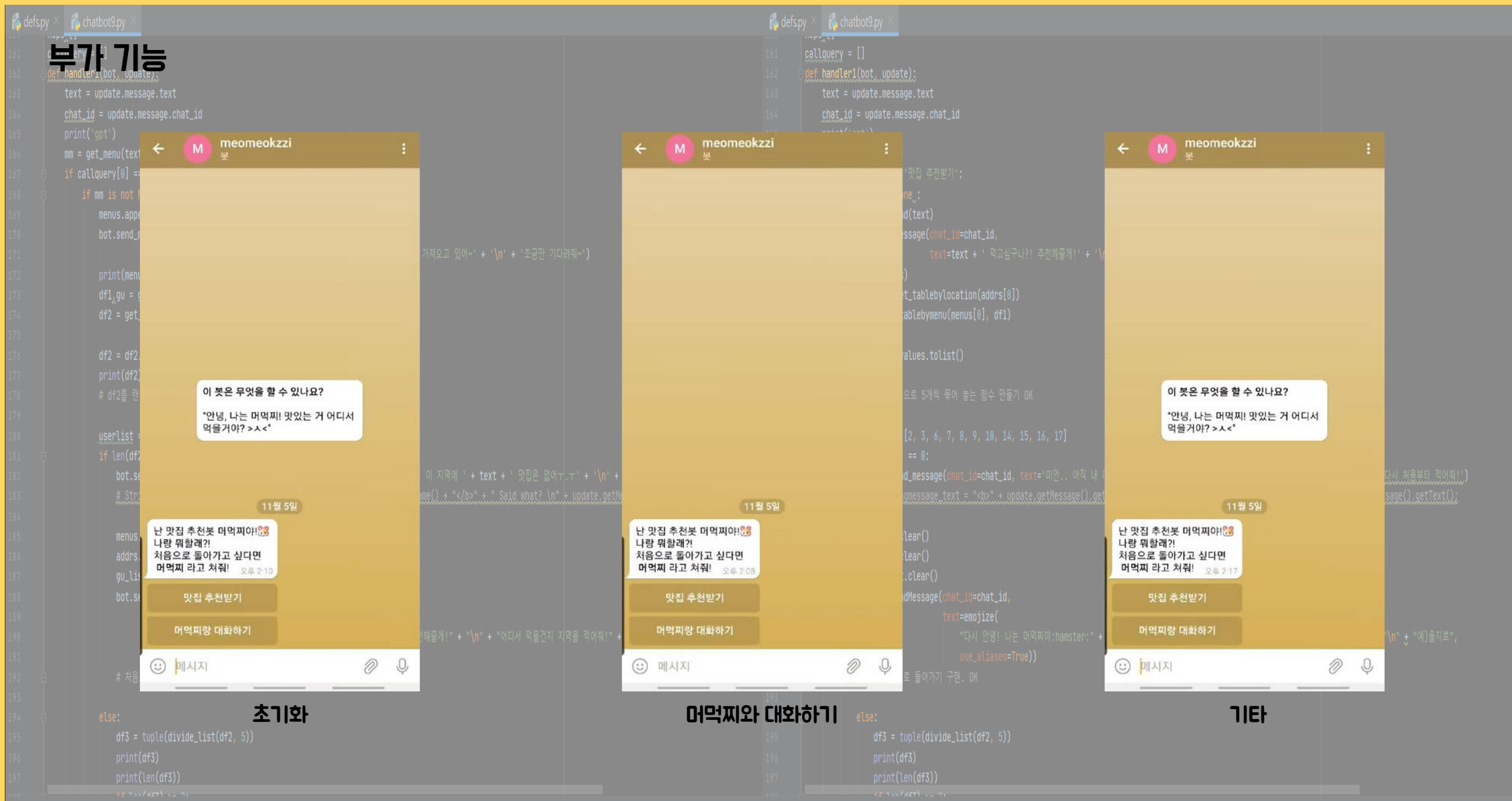
맛집 추천받기
머먹찌랑 대화하기

메시지

장소 > 날씨

4. 시연

부가 기능



4. 시연

부가 기능

이 봇은 무엇을 할 수 있나요?

"안녕, 나는 머먹찌! 맛있는 거 어디서 먹을거야? >ㅅ<"

난 맛집 추천봇 머먹찌야! 🐻
나랑 뭐할래?!
처음으로 돌아가고 싶다면
머먹찌 라고 쳐줘! 오후 2:10

초기화

11월 5일

난 맛집 추천봇 머먹찌야! 🐻
나랑 뭐할래?!
처음으로 돌아가고 싶다면
머먹찌 라고 쳐줘! 오후 2:08

오후 2:08

맛집 추천받기

머먹찌랑 대화하기

머먹찌와 대화하기

이 책은 무엇을 할 수 있나요?

"안녕, 나는 머먹찌! 맛있는 거 어디서 먹을거야? >ㅅ<"

난 맛집 추천봇 머먹찌야! 🐶
나랑 뭐할래?!

처음으로 돌아가고 싶다면
머먹찌 라고 쳐줘!

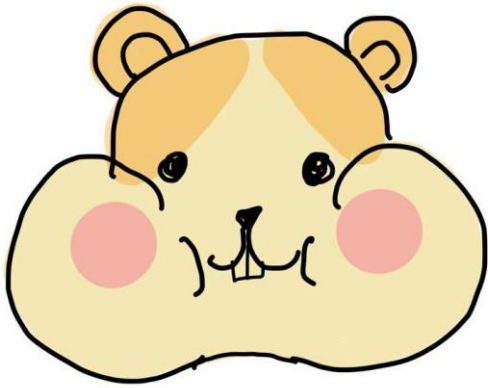
오후 2:17

기타

5. 기대 효과



5. 기대 효과



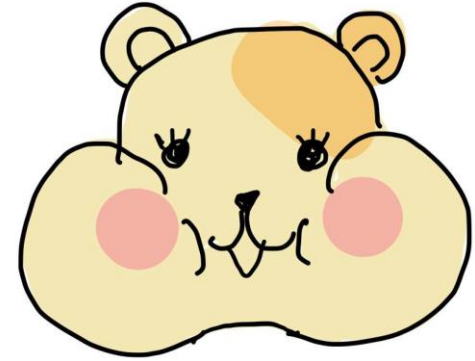
PERSONAL

기존 배달서비스 앱과는 달리
날씨, 기분을 기반으로 초개인화 된
메뉴 추천 가능



ENTERTAIN

인공지능 챗봇 기능 추가로
색다른 경험 및 즐거움 선사



UNTACT

언택트 시대에 맞춰 인공지능 챗봇
추천시스템 기반의 배달 플랫폼
및 비즈니스 생태계 구축



6. 향후 계획

6. 향후 계획

폐업 여부 확인 후 데이터 업데이트

지속적인 사용자 데이터 축적 및 학습으로 사용자 맞춤 정보 제공

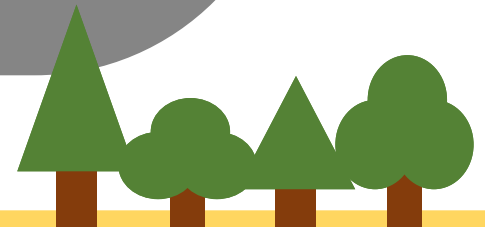
보안 및 서버 구축 후 데이터 관리

현 시간 기준 최다 주문한 메뉴 랜덤 추천

자체 어플리케이션 개발 및 배달 업체와의 제휴



7. 개발 환경



7. 개발 환경

구분		항목	세부 내용
SW개발환경	OS	Window	Window 10 64bit
	개발도구	Colab, Anaconda	Colab, Anaconda
	IDE	Jupyter notebook, Colab	Jupyter notebook, Colab
	개발언어	Python	Python 3.7(64bit)
	DB	RDB, GraphDB	Mysql, Neo4j
	서버	미정	미정
프로젝트관리환경		Google Drive	Google Drive
		추후관리 : Git-hub	추후관리 : Git-hub

감사합니다



참고 자료

명칭	내용	출처
감정분석(SA)	사용자 리뷰의 긍/부정 분류	네이버 영화리뷰 긍/부정 데이터 셋
개체명 인식	LSTM + CRF 활용	한국해양대학교 자연어처리 연구실
인공지능	koGPT2를 이용한 대화 처리	AWS, SKT
날씨 분석	공공데이터포털 동네예보 조회서비스 API	공공데이터포털

뉴스

<http://www.hani.co.kr/arti/science/future/917004.html#csidx0ab8da5abe753049a938>

67de6a8c2b5 <https://www.fnnews.com/news/202007201049197658>

<http://www.dailypop.kr/news/articleView.html?idxno=47068>

참고 문헌

스마트 기반 날씨정보 콘텐츠 개발 연구;날씨에 따른 위치기반 음식 추천 애플리케이션 개발
감정분석 기반 음식추천 어플리케이션 설계

분석 언어 및 도구

