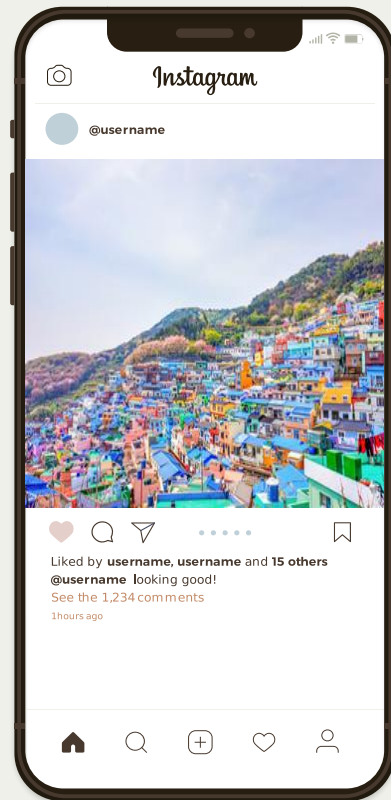


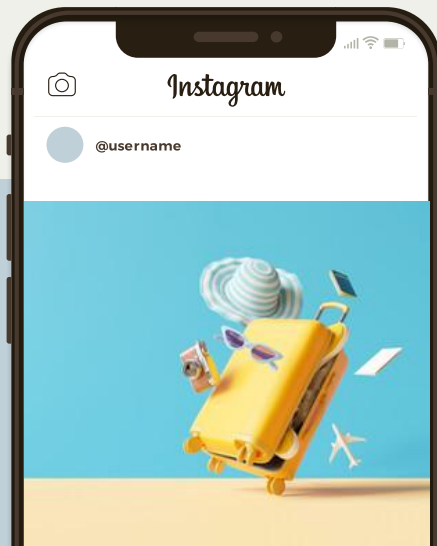
Chatbot for Travel Text_analysis

2023.06.08

김창균|송수린|송찬의|현정환



“국내 여행지 추천”



1

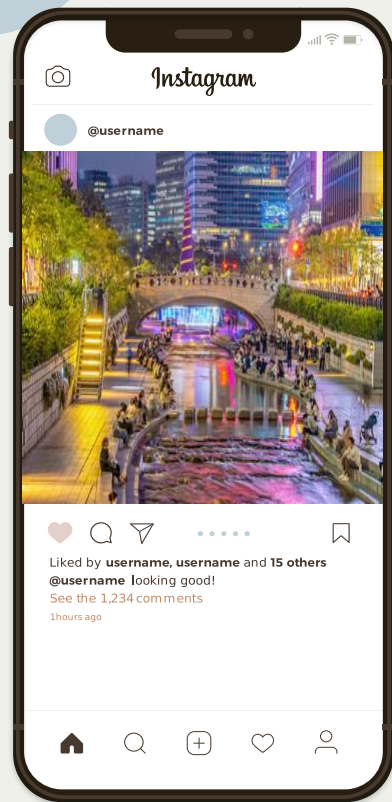
프로젝트 목적

2

프로젝트 진행 과정

3

데이터 설명



4

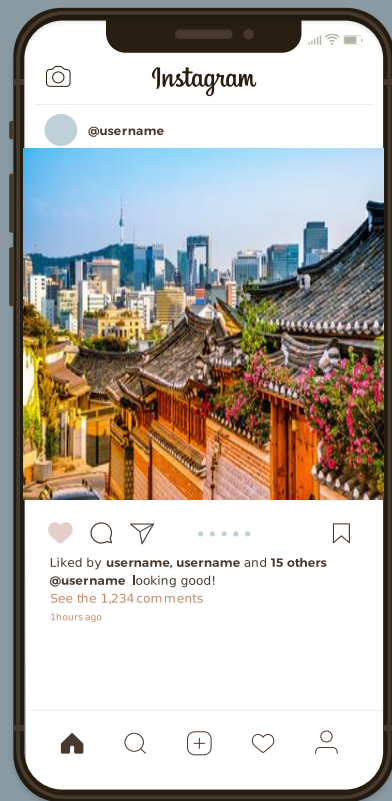
모델 설명

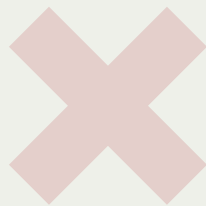
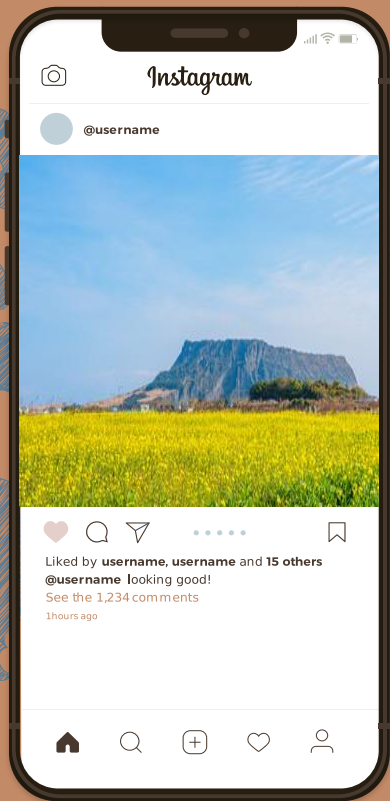
5

DB 생성

6

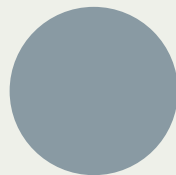
챗봇 구축





개발 목적

코로나 종식 선언 이후,
국내 여행을 계획 중인 사람들을 위한
여행지 추천 챗봇을 만들고자 하였습니다.



진행과정

데이터 수집

- 크롤링
- 다운로드

의도 파악 모델 구축

DB 생성

- 예상 질의응답 작성
- 크롤링으로 추가 답변 데이터 수집

데이터 처리

- 오타자 수정
- 데이터 라벨링
- 의도 분류

개체명 인식 모델 구축

챗봇 페이지 구성

- Django

데이터 출처

AIHub & 공공데이터포털 & 크롤링

주제별 텍스트 일상 대화 데이터

<https://www.aihub.or.kr/aihubdata/data/view.do?currMenu=115&topMenu=100&aihubDataSe=realm&dataSetSn=543>

일반 상식

<https://www.aihub.or.kr/aihubdata/data/view.do?currMenu=115&topMenu=100&aihubDataSe=realm&dataSetSn=106>

용도별 목적대화 데이터

<https://www.aihub.or.kr/aihubdata/data/view.do?currMenu=115&topMenu=100&aihubDataSe=realm&dataSetSn=544>

여행 정보 데이터셋

<https://www.aihub.or.kr/aihubdata/data/view.do?currMenu=115&topMenu=100>

행정안전부_지역별(행정동) 성별
연령별 주민등록 인구수

<https://www.data.go.kr/data/15097972/fileData.do>

네이버 지식인 여행지 추천 데이터

네이버 지식인 크롤링

국내 여행지 답변 검색

<https://travel.naver.com/domestic>

데이터 구조

주제별 텍스트 일상 대화 데이터

대화 플랫폼	비율
카카오톡	81.9%
페이스북	9.0%
인스타그램	5.4%
밴드	1.8%
네이트온	1.8%

데이터 구조

일반 상식

데이터 종류	포함 내용	구조
일반상식 지식베이스	WIKI 정보를 기반으로 한 entity, attribute, value 형태의 트리플 데이터	<ul style="list-style-type: none">-entity: 위키피디아 표제어-id: entity가 동형일 경우, 이를 구분하기 위한 값-attribute: 표제어가 가질 수 있는 정보의 속성(평균 5개 내외)-value: 위키피디아 표제어의 속성에 대응하는 값

데이터 구조

용도별 목적대화 데이터

분포	주제
주제 분포	식음료, 주거와 생활, 교통 등 20여개 주제
화행 분포	단언하기, 지시하기, 언약하기, 표현하기
대화 플랫폼 분포	온라인 커머스, 교육, 유통 등의 컨택센터
화자 분포	남성, 여성, 연령별

데이터 구조

여행 정보 데이터셋

Column명	구조	설명
type	3종류의 텍스트 데이터	여행 정보 유형 분류 구분
City	17종류의 텍스트 데이터	여행 정보 해당 도시 명
Title	텍스트 데이터	여행 정보 제목
place	텍스트 데이터	여행 정보 상세 주소

데이터 구조

행정안전부_지역별(행정동) 성별 연령별 주민등록 인구수

Column명	구조
시도명	전국 시도명 텍스트 데이터
시군구명	전국 시군구명 텍스트 데이터
읍면동명	전국 읍면동명 텍스트 데이터

데이터 수집

네이버 지식인 여행지 추천 데이터

The image displays a Jupyter Notebook on the left and a web browser on the right, illustrating the process of data collection from Naver Knowledge (Naver 지식인).

Jupyter Notebook (Left):

- File Name:** Data_to_csv.ipynb
- Code Cell 1:** Imports Selenium and sets up a Chrome driver.

```
url = "http://kin.naver.com/"
driver = webdriver.Chrome()
driver.get(url)
time.sleep(1)
```
- Code Cell 2:** Clicks the 'ico_close_layer' button and sends keys to the search input field.

```
driver.find_element(By.CLASS_NAME, 'ico_close_layer').click()
time.sleep(0.5)
driver.find_element(By.TAG_NAME, 'input').send_keys('국내 여행지 추천')
driver.find_element(By.CLASS_NAME, 'search_btn').click()
```
- Code Cell 3:** Initializes variables for title, question, and answer.

```
title = []
question = []
answer = []
```
- Code Cell 4:** A loop that iterates through search results, clicks on each result, and extracts the title, question, and answer.

```
for k in range(10):
    for j in range(1, len(driver.find_element(By.CLASS_NAME, 's_paging').find_elements(By.TAG_NAME, 'a'))):
        for i in range(len(driver.find_elements(By.CLASS_NAME, 'searchlistTitlechor'))):
            driver.find_element(By.CLASS_NAME, 'searchlistTitlechor')[i].click()
            time.sleep(1)
            driver.switch_to.window(driver.window_handles[-1])
            title.append(driver.find_element(By.CLASS_NAME, 'title').text)
            try:
                question.append(driver.find_element(By.CLASS_NAME, 'c-heading__content').text)
            except:
                question.append('')
            answer.append(driver.find_element(By.CLASS_NAME, 'c-heading__answer__content').text)
            driver.close()
            driver.switch_to.window(driver.window_handles[0])
            driver.find_element(By.CLASS_NAME, 's_paging').find_elements(By.TAG_NAME, 'a')[j].click()
            time.sleep(1)
```

Web Browser (Right):

- URL:** kin.naver.com/search/listnaver?query=국내+여행지+추천
- Page Title:** 네이버 지식인
- Search Query:** 국내 여행지 추천
- Results:** The page shows search results for '국내 여행지 추천' (Domestic Travel Recommendation). The first result is titled '전체 지식인 (1-10/11,307)' (All Knowledge (1-10/11,307)).

데이터 수집

국내 여행지 답변 검색


NAVER 여행정보 | 여행상품

마이페이지 로그인


국내여행 해외여행

#봄꽃여행 #여름 #제철여행 #심내여행지


꼭 밖에서만 놀아야 하나요?
심내여행지도 있어요 ♡




제주 여행 십지코지 아
푸아를라넷
제주도



전북 완주여행 코스 삼
레문화예술촌 심내...
완주



강원도 강릉 가볼만한
곳 심내 핫플 강릉 ...
강릉




서울 근교 여행 인천
심내 갈만한곳 송도 ...
인천

테마 더보기 >


찾고 계신 호텔이 있나요?
여행의 시작은 호텔 예약부터~

최근 30일 최저요금


제주 서울 강원 부산 여수 광주 전주 인천




라비드 아틀란 2
★ 8
61,901 원 ~




시그니엘 부산
★ 9 | 5성급
276,750 원 ~




라발스 호텔 부산
★ 8 | 4성급
100,615 원 ~



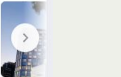
신라스테이 해운대
★ 8 | 4성급
121,232 원 ~



토요코인 부산 해운대
2호점
★ 8 | 2성급
65,179 원 ~

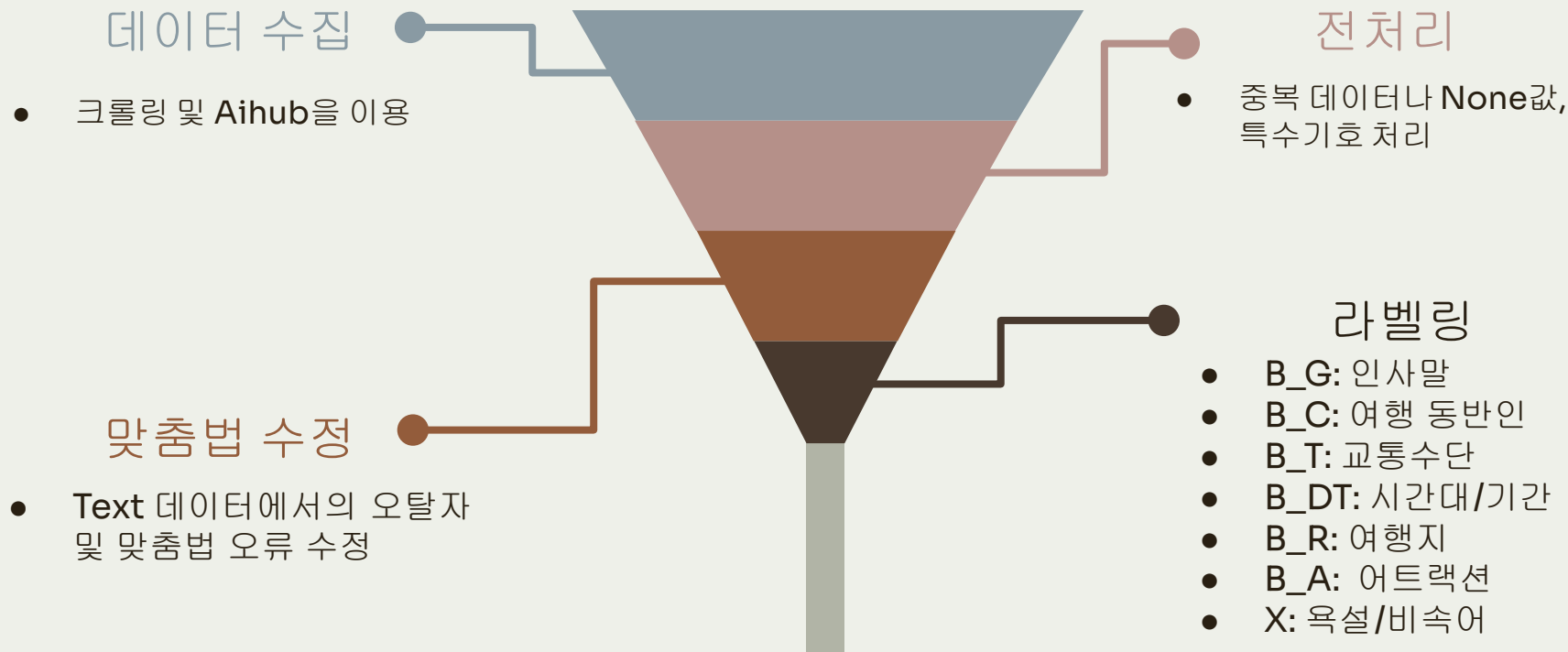


파라다이스 호텔 부산
★ 8 | 5성급
242,011 원 ~



그랜드 호텔 부산
★ 8 | 5성급
224,410 원 ~

데이터 수집 및 처리 과정



1,257,779



전처리 과정을 통해 425개 감소
총 1,257,354의 데이터 사용

의도 파악 모델

모델 구축 과정

- 형태소 분석
- 불용어 처리
- 제로패딩
- 학습용,검증용 데이터셋 생성
- 모델 학습 및 평가

모델 평가

- Accuracy 91.12%
- Loss 0.233

모델 생성

```
model = Model(inputs=input_layer, outputs=predictions)
model.compile(optimizer=adam,
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
```

모델 생성 코드

CNN 모델 정의

```
input_layer = Input(shape=(MAX_SEQ_LEN, ))
embedding_layer = Embedding(VOCAB_SIZE, EMB_SIZE,
                             input_length=MAX_SEQ_LEN)(input_layer)
dropout_emb = Dropout(rate=dropout_prob)(embedding_layer)

conv1 = Conv1D(
    filters=128,
    kernel_size=3,
    padding='valid',
    activation=tf.nn.relu)(dropout_emb)
pool1 = GlobalMaxPool1D()(conv1)

conv2 = Conv1D(
    filters=128,
    kernel_size=4,
    padding='valid',
    activation=tf.nn.relu)(dropout_emb)
pool2 = GlobalMaxPool1D()(conv2)

concat = concatenate([pool1, pool2])

hidden = Dense(128, activation=tf.nn.relu)(concat)
dropout_hidden = Dropout(rate=dropout_prob)(hidden)
logits = Dense(4, name='logits')(dropout_hidden)
predictions = Dense(4, activation=tf.nn.softmax)(logits)
adam = Adam(learning_rate=0.0001)
```

의도 파악 모델

의도 분할 과정

- 의도 클래스 별 레이블 설정
- 의도 분류 모델 불러오기
- 형태소 분석
- 문장내 키워드 추출(불용어 제거)
- 패딩처리
- 의도 클래스 예측

의도 파악 모델 코드

```
class IntentModel:
    def __init__(self, model_name, preprocess):
        # 의도 클래스 별 레이블
        self.labels = {0: "추천", 1: "예약", 2: "정보", 3: "기타"}

        # 의도 분류 모델 불러오기
        self.model = load_model(model_name)

        # 챗봇 Preprocess 객체
        self.p = preprocess

    # 의도 클래스 예측
    def predict_class(self, query):

        # 형태소 분석
        pos = self.p.pos(query)

        # 문장내 키워드 추출(불용어 제거)
        keywords = self.p.get_keywords(pos, without_tag=True)
        sequences = [self.p.get_wordidx_sequence(keywords)]

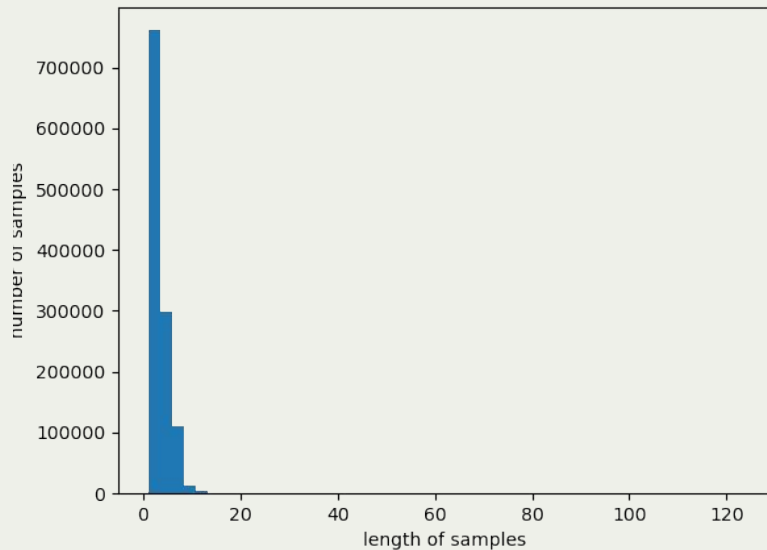
        # 패딩처리
        padded_seqs = preprocessing.sequence.pad_sequences(sequences,
            maxlen=MAX_SEQ_LEN, padding='post')
        predict = self.model.predict(padded_seqs)
        predict_class = tf.math.argmax(predict, axis=1)
        return predict_class.numpy()[0]
```

객체명 인식 모델

모델 구축 과정

- 테스트 데이터 한 문장의 단어를 한 리스트에 저장
- 단어와 태그 연결
- 샘플 단어 시퀀스 길이를 고려하여 **max_len** 설정
- 말뭉치 데이터에서 단어와 **BIO**태그만 불러와 학습용 데이터셋 생성
- 토큰나이저 정의
- 학습용 단어 시퀀스 생성 및 패딩처리
- 학습, 테스트 데이터 분리 및 원핫인코딩
- 모델 정의 및 평가

단어 시퀀스 길이



개체명 인식 모델

모델 구축

모델 정의 (Bi-LSTM)

```
model = Sequential()
model.add(Embedding(input_dim=vocab_size, output_dim=256, input_length=max_len, mask_zero=True))
model.add(Bidirectional(LSTM(256, return_sequences=True, dropout=0.40, recurrent_dropout=0.25)))
model.add(TimeDistributed(Dense(tag_size, activation='softmax')))
model.compile(loss='categorical_crossentropy', optimizer=Adam(0.001), metrics=['accuracy'])
model.fit(x_train, y_train, batch_size=128, epochs=10)
```

개체명 인식 모델 클래스 지정

```
class NerModel:
    def __init__(self, model_name, preprocess):
        # BIO 태그 클래스별 레이블
        self.index_to_ner = {1: 'O', 2: 'B_DT', 3: 'B_C', 4: 'B_R', 5: 'B_G', 6: 'B_A', 7: 'B_P', 8: 'B_T', 9: 'X', 10: 'I', 0: 'PAD'}

        # 개체명 인식 모델 불러오기
        self.model = load_model(model_name)

        # 챗봇 Preprocess 객체
        self.p = preprocess
```

Ner tag	cnt
B_A	40131
B_C	108264
B_DT	155999
B_G	46214
B_P	18582
B_R	59293
B_T	14618
I	9484
O	3367902
X	14424

개체명 인식 모델

모델 평가

- 모델 정확도
Accuracy 99.85%
Loss 0.0012
- F1 모델 정확도
F1-score 99.4%

f1 스코어

```
from seqeval.metrics import f1_score, classification_report
y_predicted = model.predict([x_test]) # 테스트 데이터셋의 NER 예측
pred_tags = sequences_to_tag(y_predicted) # 예측된 NER
test_tags = sequences_to_tag(y_test) # 실제 NER
print(classification_report(test_tags, pred_tags))
print('F1-score: {:.1%}'.format(f1_score(test_tags, pred_tags)))
```

F1-score

	precision	recall	f1-score
-	1.00	1.00	1.00
_A	0.99	0.96	0.98
_C	1.00	1.00	1.00
_DT	1.00	0.99	0.99
_G	1.00	1.00	1.00
_P	0.99	0.96	0.98
_R	1.00	0.99	0.99
_T	1.00	0.99	0.99
F1-score	99.4%		

DB 생성

데이터 베이스 내용 및 생성 코드

1. Chatbot_train_data

```
CREATE TABLE chatbot_train_data (  
  id INT PRIMARY KEY AUTO_INCREMENT NOT NULL,  
  intent VARCHAR(45),  
  ner VARCHAR(45),  
  QUERY TEXT,  
  answer TEXT NOT NULL  
);
```

2. Travel_chat

```
class Chat(models.Model):  
  idx = models.AutoField(primary_key=True)  
  query = models.CharField(max_length=500, null=False)  
  answer = models.CharField(max_length=1000, null=False)  
  intent = models.CharField(max_length=50, null=False)
```

데이터 베이스 내용

1. Chatbot_train_data - 예상 질문과 답변 검색

id	intent	ner	QUERY	answer
1	기타	B_G	안녕하세요	네 안녕하세요 :D 반갑습니다. 저는 챗봇입니다.
2	기타	B_G	반가워요	네 안녕하세요 :D 반갑습니다. 저는 챗봇입니다.
3	추천	B_R	{B_R}로 놀러가고 싶어요	네, {B_R}로 가고 싶으시군요!
4	추천	B_C	{B_C}항 놀러갈만한 데 어디 있을까요	네, {B_C}와 갈 만한 곳을 찾고 계시군요!
5	추천	B_R,B_A	{B_R}에 {B_A}는 어떤 게 있을까요	네, {B_R}에 있는 {B_A}를 찾고 계시군요!
6	추천	B_A	{B_A}가 좋은 곳 어디 있을까요	네, {B_A}가 좋은 곳을 찾고 계시군요!
7	추천	B_R,B_A	{B_R}에 {B_A} 추천해주세요	네, {B_R}에 가볼만한 {B_A}를 찾고 계시군요!
8	추천	B_DT,B_C	{B_C}와 {B_DT}에 가기 좋은 곳은 어디예요	네, {B_DT}에 {B_C}와 가기 좋은 곳을 찾고 계시군요!
9	추천	B_DT	{B_DT}에 갈 만한 곳이 있을까요	네, {B_DT}에 가기 좋은 곳을 찾고 계시군요!
10	추천	B_T	{B_T}를 타고 갈건데 어디로 가는게 좋을까요?	네, {B_T} 수단을 이용하여 갈 곳을 찾고 계시군요!
11	추천	B_DT	{B_DT}에 놀러가고려고 해요	네, {B_DT}에 여행 계획을 세우고 계시군요!
12	추천	B_DT,B_A	{B_DT}에 {B_A} 할 만한 곳 있을까요?	네, {B_DT}에 {B_A} 할 곳을 찾고 계시군요!
13	정보	B_DT,B_R,B_A	{B_DT}에 {B_R} 가려고 하는데 {B_A} 추천해주세요.	네, {B_DT}에 {B_R}에서의 {B_A}를 찾고 계시군요!
14	추천	(NULL)	여행지 추천해주세요	네, 혹시 가고 싶은 지역이나, 일정 계획 등을 알려주실 ...

2. Travel_chat - 챗봇 화면에 출력

idx	query	answer	intent
194	안녕하세요	네 안녕하세요 :D 반갑습니다. 저는 챗봇입니다.	기타
195	부산으로 놀러가고 싶은데 어디로 가면 좋을까요?	네, 부산(으)로 가고 싶으시군요! 대전천누리길, 담안골...	추천
196	신혼여행지 추천해주세요	죄송합니다. 질문 내용을 이해하지 못했습니다.	추천
197	신혼여행지 추천해주세요	추천하는 여행지로는 제주특별자치도, 부산광역시, 대구...	추천

□
□
□

DB 생성

데이터 베이스 내용 및 생성 코드

3. Detail_Ans_DB

DB연결

```
cnx = MySQLdb.connect(user='web', passwd='1234', host='localhost', db='travel')
```

```
cursor = cnx.cursor()
table_name = 'total_attraction'
```

테이블 생성

```
create_table_query = f"CREATE TABLE IF NOT EXISTS {table_name} ({', '.join(['{col}' for col in df.columns])})"
cursor.execute(create_table_query)
```

크롤링 데이터 입력

```
for row in df.itertuples():
    insert_query = f"INSERT INTO {table_name} VALUES {tuple(row[1:])}"
    cursor.execute(insert_query)
```

Commit

```
cnx.commit()
```

데이터 베이스 내용

3. Detain_Ans_DB - 질문에 포함된 개체명 검색 후 크롤링한 데이터

시도명	시군구명	유연동명	어트랙션	어트랙션_목록
서울특별시	종로구	정운효자동	여협지	경복궁, 광화문광장, 수성동계곡, 서촌마을, 광화문, 세...
서울특별시	종로구	정운효자동	불거리/불거리/명소	대한민국역사박물관, 세종문화회관, 광화문광장, 국립...
서울특별시	종로구	정운효자동	취미생활	로라홀, 커스텀테라피, 가회민화박물관, 동림매듭공방, ...
서울특별시	종로구	정운효자동	맛집	아끼니구소항 경복궁점, 서촌급상고로케, 잡박진메일 서...
서울특별시	종로구	정운효자동	카페	서촌급상고로케, 파스텔커피옥스 서촌점, 더숲 초소책방...
서울특별시	종로구	정운효자동	아이와함께	대한민국역사박물관, 국립고궁박물관, 운동주문학관, ...
서울특별시	종로구	사직동	여협지	돈의문박물관마을, 경희궁, 사직단, 세종마을을식문화...
서울특별시	종로구	사직동	불거리/불거리/명소	그라운드시스 서촌, 서울역사박물관, 통의도 보안여관, ...
서울특별시	종로구	사직동	취미생활	초록만생활연구소, 로라홀, 커스텀테라피

□

□

□

충청북도	단양군	가곡면	맛집	신마구 서충수협, 송암탈조가집, 송암탈박곡수, 남한강...
충청북도	단양군	가곡면	카페	탈림카페, 카페농소와, 구름, 모듬이, 커피엘로스, 하삼...
충청북도	단양군	가곡면	아이와함께	목계술방, 충주고구려천문과학관, 리틀비틀 서충주신도...
충청북도	단양군	영춘면	여협지	온달관광지, 구인사, 소백산 자연휴양림, 온달동굴, 온달...
충청북도	단양군	영춘면	불거리/불거리/명소	구인사, 소백산 자연휴양림, 온달관광지, 온달동굴, 리틀...
충청북도	단양군	영춘면	취미생활	미육공방
충청북도	단양군	영춘면	맛집	리틀포레스트, 장미식당, 해성한식, 시즈부, 산촌가든, ...
충청북도	단양군	영춘면	카페	리틀포레스트, 시즈부, 카페, 온, 황금물결 카페, 이 장소...
충청북도	단양군	영춘면	아이와함께	온달관광지, 불교전통중앙박물관, 단양다누리아쿠아리...
충청북도	단양군	어상천면	여협지	느티나무학교, 일광굴, 네덜란드풍차마을, 느티나무학...
충청북도	단양군	어상천면	불거리/불거리/명소	문수사, 낙원식당, 단양무지개오토캠핑장, 가마실 오토...

Chatbot for Travel

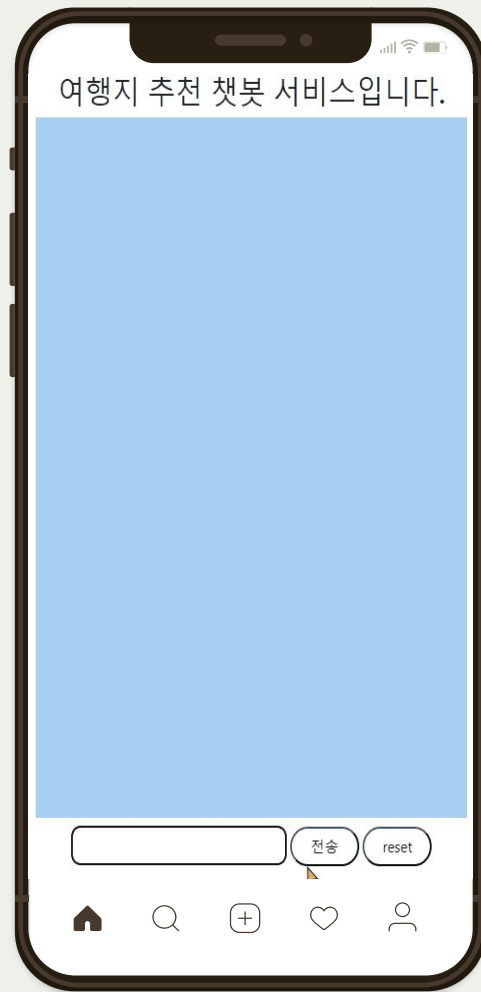
- **Django**를 사용하여 대화형 챗봇 화면 구축

- 답변 가져오기 - **DB**이용

개체명이 포함되지 않은 질문(인사말 및 비속어)의 경우

개체명을 인식하여 해당 질문과 연결된 DB상의 답변 불러오기

개체명에 알맞는 값 대입 후 화면에 띄울 DB에 추가



Chatbot for Travel

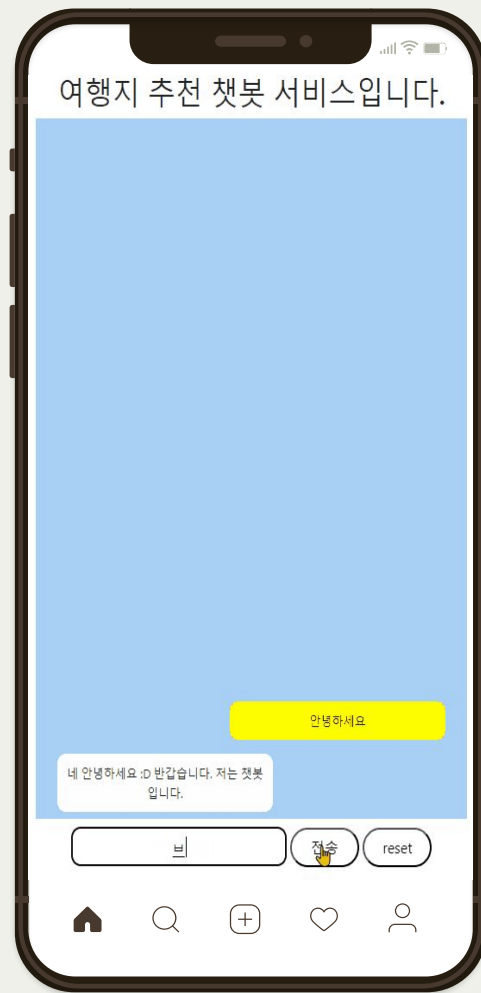
- **Django**를 사용하여 대화형 챗봇 화면 구축
 - 답변 가져오기 - 네이버 검색하여 태그 크롤링

개체명이 포함된 질문의 경우

질문을 분해하여 라벨링

예시) 부산 여행지 추천 = 부산 B_R / 여행지 B_A / 추천 O

앞선 DB(chatbot_train_data) 답변 +
네이버 검색 결과를 크롤링한 데이터(Detail_Ans_DB)
화면에 띄울 DB에 저장 후 출력



Chatbot for Travel

- **Django**를 사용하여 대화형 챗봇 화면 구축
- 답변 가져오기 - 네이버 여행지 추천 사이트 크롤링

검색 결과가 나오지 않는 지역의 경우

ADD버튼을 누르면 질문으로 받은 개체명을 네이버 여행 추천 사이트 내에서 검색하여 정보 크롤링

화면에 띄울 DB에 저장 후 출력



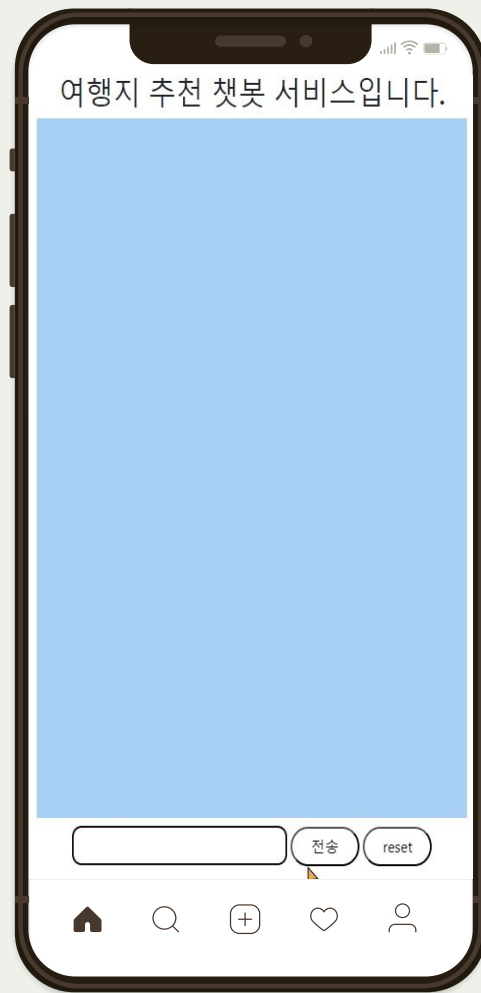
Chatbot for Travel

- Django를 사용하여 대화형 챗봇 화면 구축

- 화면에 질문과 답변 출력

화면에 띄울 DB(travel_chat)에 저장된 질문과 답변을 출력

idx	query	answer	intent
194	안녕하세요 1	네 안녕하세요 :D 반갑습니다. 저는 챗봇입니다. 2	기타
195	부산으로 놀러가고 싶은데 어디로 가면 좋을까요? 3	네, 부산(으)로 가고 싶으시군요! 대전천누리길, 담안골... 4	추천
196	신혼여행지 추천해주세요 5	죄송합니다. 질문 내용을 이해하지 못했습니다. 6	추천
197	신혼여행지 추천해주세요 7	추천하는 여행지로는 제주특별자치도, 부산광역시, 대구... 8	추천



역할 분담



향후 개선 과제 및 느낀점

향후 개선 과제

- 모델 구축
 - BIO 태깅 보완
 - 추가 여행지 업데이트
- 챗봇 구축
 - 인터페이스 개선
 - 로그인 기능 추가
 - DB 자동 리셋 기능 추가
 - 예약 시스템 구축
 - 예상 질문 추가 및 세분화

1

김창균 


어떻게든 챗봇을 구현할 수 있어서 좋았고, 중간중간에 허점과 오류가 발견되었지만 시간 부족으로 인해 개선하지 못한 것이 아쉬웠다.

2

현정환 


이번 프로젝트에서 텍스트 분석과 챗봇에 대해 많이 배울 수 있었다. 다만 시간이 충분했다면 답변에 대한 DB를 단순한 크롤링보다 크롤링한 데이터를 다시 분석해서 빈도수나 중요 키워드를 위주로 답변을 만들어서 보다 좋은 DB를 구축할 수 있었을 것 같았지만 그러지 못한 점이 아쉽다.

3

송수린 

크롤링을 통해 보다 다양한 답변을 보여줄 수 있었지만, 검색 과정에서 동일한 이름의 지역명으로 인한 잘못된 답변을 방지하지 못한 것이 아쉬웠다.

4

송찬의 

라벨링 과정에서 몇개의 데이터를 제외한 나머지 데이터들은 직접 입력하다보니 라벨링에서 부족한 부분이 있어서 아쉽고, 로그인 기능을 넣지 못해서 사용자별 DB가 아니라 같은 DB화면을 공유해서 아쉽다.

Thanks!

DO YOU HAVE ANY QUESTIONS?

thdcksdml980@naver.com

+02 010 3938 1049

naver.com



CREDITS: This presentation template was created by KHS2, and includes icons by Slidesgo, and infographics & images by Adobe Stock

2023-06-08