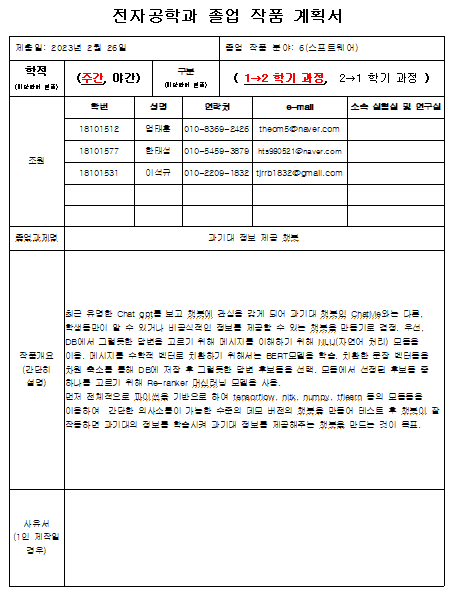
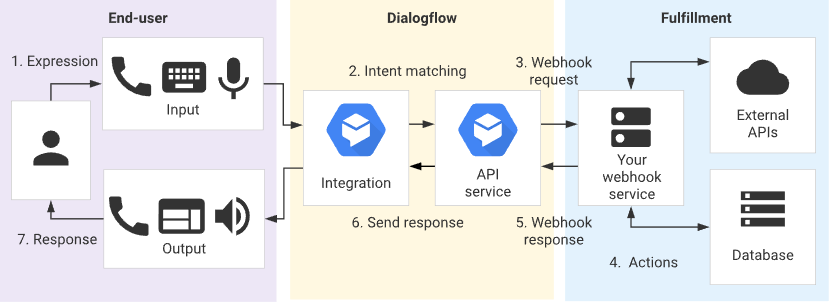
2주차 : 2023.02.28 – 2023.03.06

Python 을 활용한 chat-bot을 만들고자 제안서를 제출했다.



3주차 : 2023.03.07 – 2023.03.13

# 1. Dialogflow

- 자연어처리 모듈로 구글 클라우드에서 서비스 중인 dialogflow를 사용하기로 결정했다.

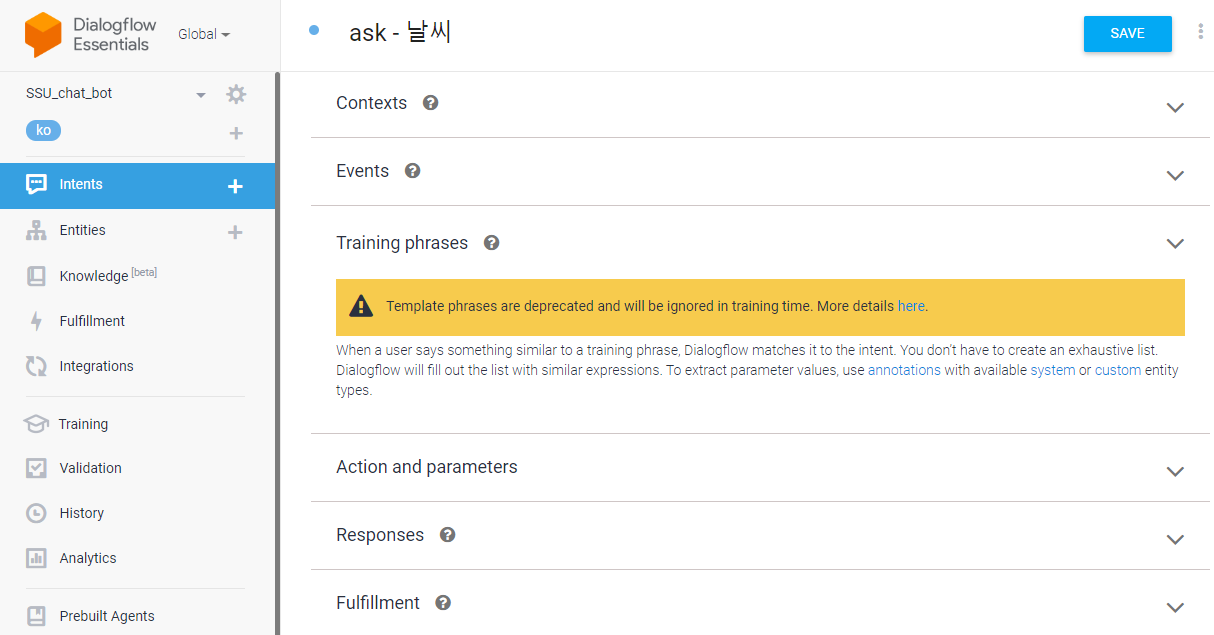
(그림1.) dialogflow fullfillment

- 예상대는 사용법

* 최종 사용자가 표현을 입력하거나 말합니다.
* Dialogflow가 최종 사용자 표현을 인텐트와 일치시키고 매개변수를 추출합니다.
* Dialogflow가 웹훅 서비스에 [웹훅 요청](https://cloud.google.com/dialogflow/docs/fulfillment-webhook?hl=ko" \l "webhook_request) 메시지를 보냅니다. 이 메시지에는 일치하는 인텐트, 작업, 매개변수, 인텐트에 정의된 응답에 대한 정보가 포함됩니다.
* 웹훅 서비스에서 제공한 응답을 DIalogflow가 최종 사용자에게 보냅니다.

출처 : (<https://cloud.google.com/dialogflow/es/docs/fulfillment-overview?hl=ko>)

- 실제 사용화면



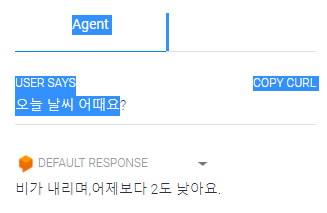
(그림2.) dialogflow intent 추가 화면

* Context : 전후 대화의 문맥을 파악하여 대화 구성
* Events : 쳇봇이 의도 파악을 위해 최종 사용자에게 질문을 던졌을 때 분기점
* Training phrase : intent에 맞는 예상되는 질문입력 자체적 학습 후 입력하지 않은 구문에도 반응

Ex) intent : 날씨질문

Training phrase : 날씨, 날씨 어때, 오늘 날씨 어떤가요?

* Response : 정해진 답변 답변가능
* fullfillment 외부 사이트와의 연결 (웹훅)



(그림3.) dialogflow 러닝 후 대화시도

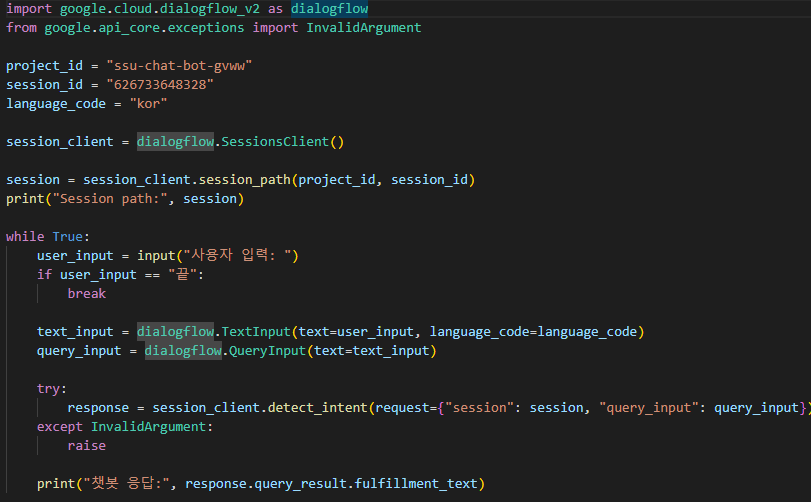




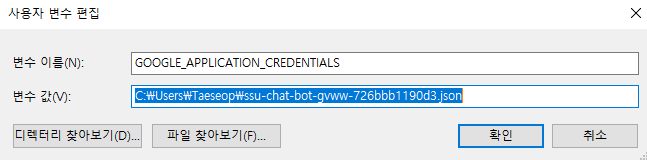
(그림4.) dialogflow 학습 전 후

## 2. Python과 연동

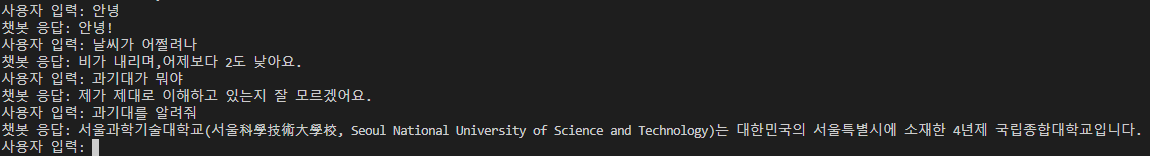
- web상에서 동작하는 dialogflow에서 key를 받아와 python에서 동작하게 코드를 짜보았다.



(그림 5.) python code



(그림 6.) 환경 변수 설정



(그림 7.) 환경 변수 설정

4주차 : 2023.03.14 – 2023.03.20

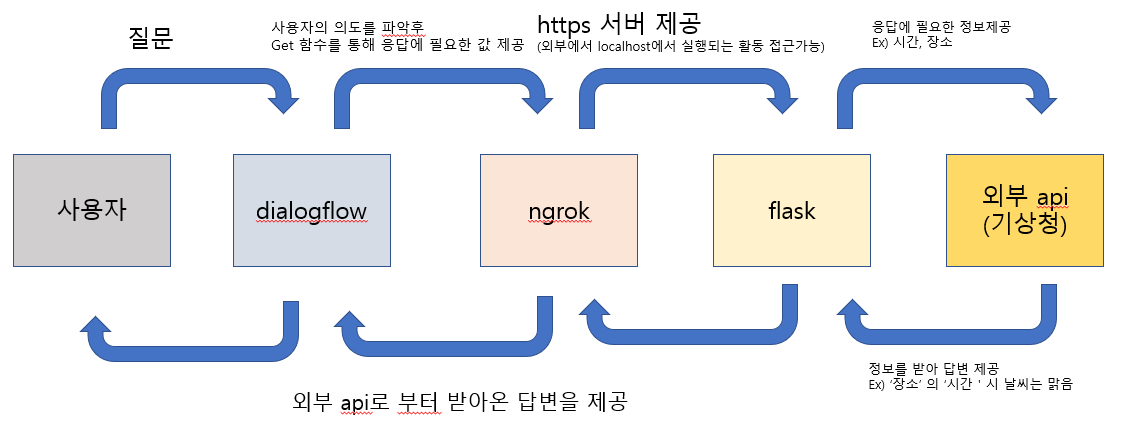
진행중인 목표

1. dialogflow webhook 기능 사용하여 외부 api(기상청)와 연동하기

2. 크롤링한 정보와 사용자와 이야기한 정보를 저장하기 위해 MySQL서버 구축 및 연동

3. dialogflow 사용범위 (교수님 피드백)

# 1.dialogflow webhook 으로 기상청과 연동하기



( 그림 1.) 원하는 시스템 구조

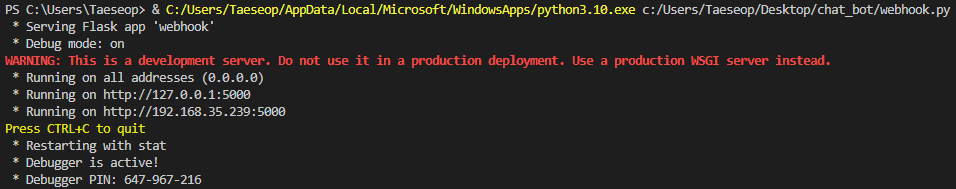
Dialogflow 만으론 현재시간의 날씨를 대답하는등의 활동에는 무리가 있기 때문에 외부 api 와의 연동이 필수적으로 필요하다. Dialgflow 자체적으로 지원하는 Webhook 기능을 통해 기상청에서 현재 날씨정보를 실시간으로 가져와 사용자에게 정보를 제공하는 서버와 연동하였다.

**■ flask 를 통한 내부 localhost 만들기**

**기상청 api와 dialog의 연결역활을 할 서버를 만드는 tool로 flask를 사용하였다.**

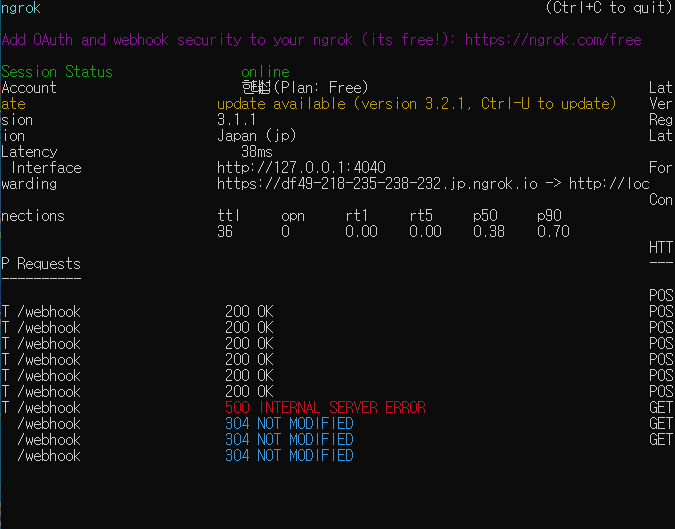
코드: <https://blog.naver.com/hts980521/223057400237>

( 그림 2.) 기본적인 응답을 보내는 flask 서버 동작 code



( 그림 3.) code 작동시

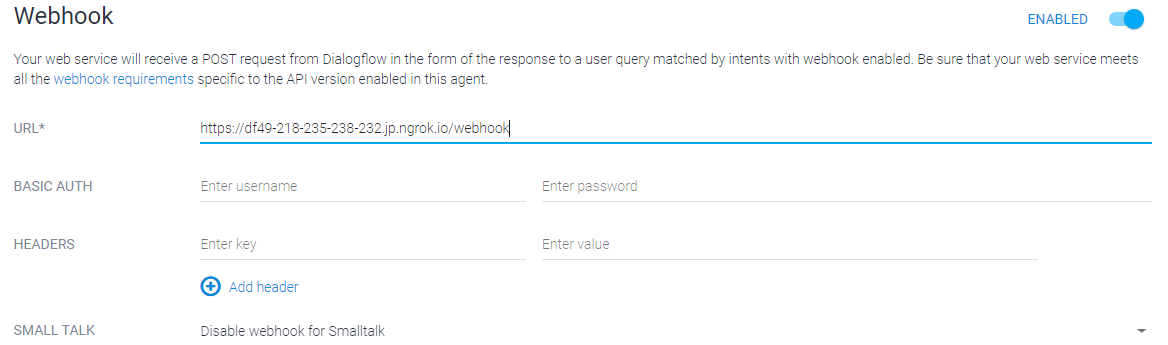
**■ ngrok 을 사용하여 외부에서 localhost 접근할 수 있도록 만들어주기**



( 그림 4.) ngrok을 통해 https 서버를 받는다

webhook을 위한 기본적인 틀만 잡은 flask로써 port 5000의 내부서버에 생성된 서버이다. 하지만 http인 즉 localhost주소는 dialogflow와 연동이 불가능하기 때문에 외부에서 접근이 가능하도록 만들어주는 https 서버를 제공해주는 프로그램인 ngrok을 같이 사용해야한다.

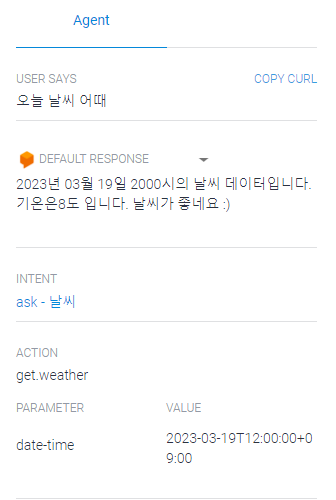
**■dialogflow webhook하기**

****

( 그림 5.) webhook

**■ 날씨를 묻는 질문이 나왔을시 , 기상청 api로부터 현재 노원구 날씨를 불러오는 코드 작성**

**코딩,출처: (**<https://blog.naver.com/hts980521/223057258051>)



( 그림 6.) 정상작동 하는 모습

# 2. 크롤링한 정보와 사용자와 이야기한 정보를 저장하기 위해 MySQL서버 구축 및 연동

2-1) 크롤링 모듈

: requests와 beutifulsoup 라이브러리를 이용해 쿼리를 형성

Mysql connector라이브러리를 이용해 서버에 있는 테이블에 자동으로 정보가 들어가도록 설계와

SQLAlchemy 라이브러리를 이용해 Session 객체를 형성하여 스타트

Datetime 라이브러리를 이용해 실시간 시간 정보를 불러옴

import requests

from bs4 import BeautifulSoup

from sqlalchemy.orm import sessionmaker

from table import Base, Facility, Dining, Event, Other, engine

from datetime import datetime

# Session 객체 생성

Session = sessionmaker(bind=engine)

def crawl\_and\_save\_data():

# 데이터 크롤링

response = requests.get("https://www.seoultech.ac.kr")

soup = BeautifulSoup(response.content, "html.parser")

# MySQL에 데이터 삽입

session = Session()

for item in soup.find\_all("div", class\_="item"):

name = item.find("h2").text.strip()

description = item.find("p").text.strip()

if "식당" in name:

meal, menu = description.split(" - ")

dining = Dining(date=datetime.now(), meal=meal, menu=menu)

session.add(dining)

elif "시설" in name:

facility = Facility(name=name, location=location, description=description)

session.add(facility)

elif "행사" in name:

title, date\_time, location = description.split(" / ")

start\_date, end\_date = date\_time.split("~")

event = Event(title=title, start\_date=datetime.strptime(start\_date.strip(), '%Y.%m.%d %H:%M'), end\_date=datetime.strptime(end\_date.strip(), '%Y.%m.%d %H:%M'), location=location)

session.add(event)

session.commit()

session.close()

print("Data crawled and saved successfully!")

2-2) 데이터베이스 모듈

MySQL connector를 다운로드 후 파이썬에서 커넥터 라이브러리를 불러와

MySQL 서버의 host, 사용자이름, 비밀번호, 데이터베이스 이름을 입력하여   
사전에 만든 MySQL 데이터베이스와 연동

import mysql.connector

def create\_database():

try:

# MySQL 연결

conn = mysql.connector.connect(

host="localhost",

user="root",

password="sksjsm306396!",

database="database\_chatbot\_sample\_1"

)

# 데이터베이스 생성

cursor = conn.cursor()

cursor.execute("CREATE DATABASE IF NOT EXISTS database\_chatbot\_sample\_1")

# 연결 종료

cursor.close()

conn.close()

print("Database created successfully!")

except mysql.connector.Error as error:

print("Failed to create database: {}".format(error))

2-3) 테이블 모듈

SQLAlchmy 모듈을 이용해 데이터베이스를 불러오고

해당 데이터베이스에 테이블에 따라 정보를 정리하고 각 정보에 여러 구성 요소를 가진 metadata를 저장(현재 학교시설, 식당정보, 학사정보 이렇게 3개의 테이블을 구성하였고 차후 점차 테이블의 수를 늘려가도록 하겠습니다)

from sqlalchemy import create\_engine, Column, Integer, String, DateTime, Text

from sqlalchemy.ext.declarative import declarative\_base

from sqlalchemy.orm import relationship

# 데이터베이스 연결 정보

user = 'root'

password = 'sksjsm306396!'

host = 'localhost'

database = 'database\_chatbot\_sample\_1'

engine = create\_engine(f'mysql+pymysql://{user}:{password}@{host}/{database}', echo=True)

Base = declarative\_base()

class Facility(Base):

\_\_tablename\_\_ = 'facilities'

id = Column(Integer, primary\_key=True)

name = Column(String(255))

location = Column(String(255))

description = Column(Text)

class Dining(Base):

\_\_tablename\_\_ = 'dining'

id = Column(Integer, primary\_key=True)

date = Column(DateTime)

meal = Column(String(255))

menu = Column(Text)

class Event(Base):

\_\_tablename\_\_ = 'events'

id = Column(Integer, primary\_key=True)

title = Column(String(255))

start\_date = Column(DateTime)

end\_date = Column(DateTime)

location = Column(String(255))

description = Column(Text)

class Other(Base):

\_\_tablename\_\_ = 'others'

id = Column(Integer, primary\_key=True)

category = Column(String(255))

data = Column(Text)

2-4) main 모듈

import database

import table

import crawler

# 데이터베이스 생성

database.create\_database()

# 테이블 생성

table.create\_table()

# 데이터 크롤링 및 저장

crawler.crawl\_and\_save\_data()

# 데이터베이스 연결 종료

engine.dispose()

2-1), 2-2), 2-3)에서 형성한 각각의 모듈을 불러와 데이터베이스에 크롤링한 데이터가 저장된후 종료되는 코딩을 작성

\* Dialogflow에서 자체적으로 log를 제공해주어 MySQL 서버를 활용하는 방법은 취소하였습니다. 정적인 정보를 제공하는 데에는 json파일만으로 충분할 것 같아 우선 간단한 챗봇을 구현한 후 동적인 정보에 관한 처리를 어떻게 할지 고민하기로 결정하였습니다.

.5주차 : 2023.03.21 – 2023.03.27

# 목차

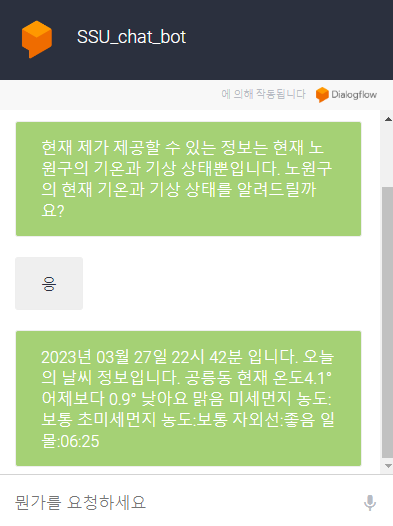
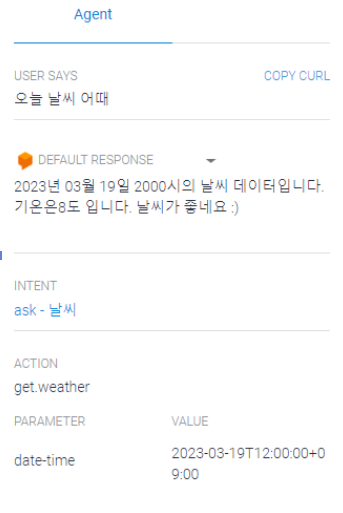
1. 날씨 데이터 : 기상청 api 연동 -> 네이버 크롤링 데이터로 변환

2. 학식 : 과기대 홈페이지에서 크롤링하여 정보 제공

3. 코드저장 blog 개설 , mysql 사용 계획 철회

## 1. 날씨 데이터

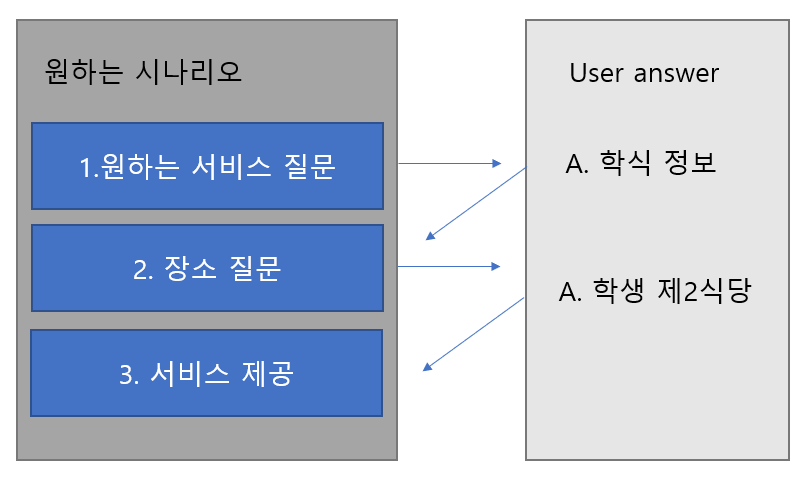
기상청 api 에서 데이터를 직접 가져다 쓰는만큼 코드가 복잡하고 길어져 주는 정보량에 비해 코드가 길다고 판단와중 크롤링을 공부하던 조원의 도움으로 naver 날씨 url을 긁어와 그대로 보여주는 코드 작성



( 그림 1.) 바꾼 날씨 응답

코딩, 출처 : (<https://blog.naver.com/hts980521/223057276695>)

## 2. 학식 데이터 크롤링

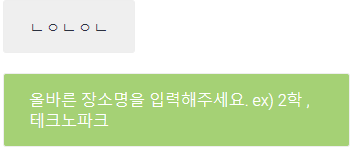


과기대 안 학식을 제공하는 건물이 총 5곳 임으로 학식 정보를 원하는 장소를 물어본 후 그곳의 정보를 제공하는 서비스를 작성했다.

**2-1) fall back intent**



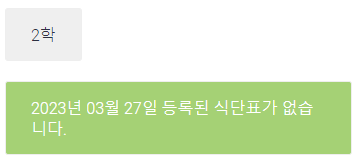
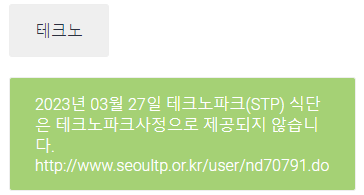
사용자의 의도를 파악하기위해 더하기 질문중 사용자가 잘못된 대답을 하였을 상황을 가정하여 fallback intent 를 추가해주었다.



**2-2) 올바른 질문이 들어온 경우**

코드: (<https://blog.naver.com/hts980521/223057360961>)

entity로 나뉘어진 parameter 값을 받아 그에 알맞은 답변을 제공해주는 코드를 작성하였다.



**2-3) 한계점**

1. 동적 웹페이지 크롤링?

지금 우리학교에서 제공하고 있는 사이트는 네이버 날씨를 알려주던 정적 웹페이지가 아니라 동적 웹페이지이기 때문에 웹페이지의 내용이 바뀔시 답변이 정상적으로 출력되지 않을것으로 추정중이다.

2. 기숙사의 존재

처음엔 제2 학생식당과 테크노파크에서만 학식이 존재하는줄알고 두곳의 data만 크롤링 하였으나, 3곳의 기숙사 역시 학식을 제공하는것을 확인하였고 추가해야한다.

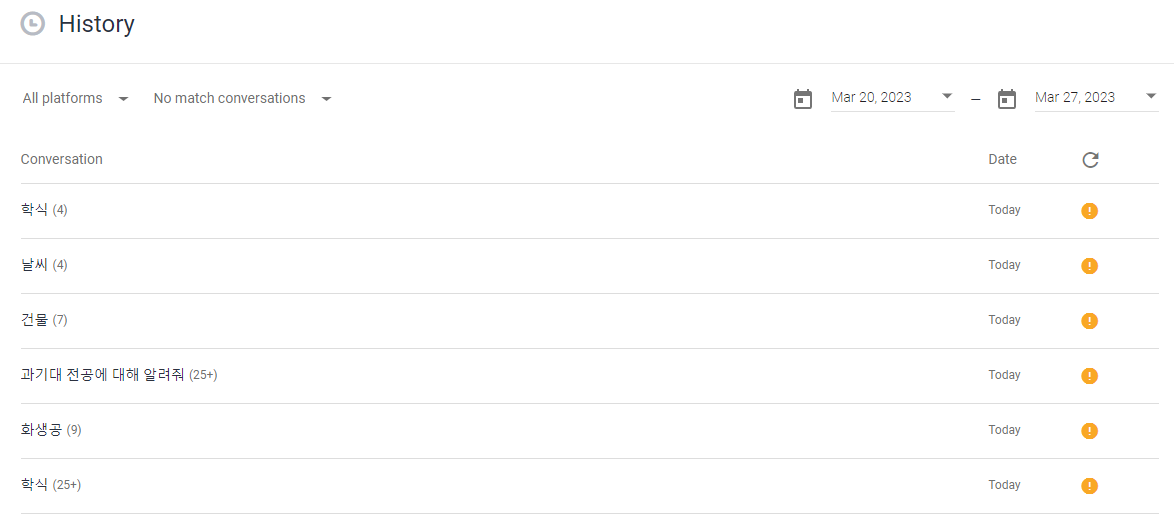
## 3. 코드저장 blog 개설 , mysql 사용 계획 철회

**3-1) 코드저장용 blog 개설**

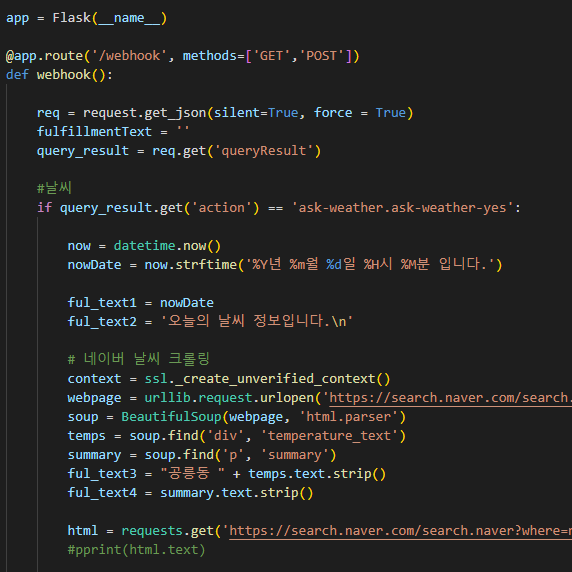
보고서에 코딩을 늘여서 적어놓는거보다 효율적으로 저장 및 설명을 적어 놓기 위해 blog를 만들었다.

**3-2) mysql 사용 계획 철회**

사용자의 질문에 챗봇이 대답 못하였을때 사용자의 질문을 저장하는 기능을 수행과 각종 사이트에서 크롤링한 데이터를 보기 좋게 정리 하기위해 mysql을 사용해보려 했으나, 대체 프로그램의 발견 및 필요성의 부제를 느끼게 되어 철회하게 되었다.



( 그림 2.) dialogflow에서 지원하는 history



( 그림 3.) python 주석 기능으로 나눈 크롤링

6주차 : 2023.03.28– 2023.04.03

## 명확한 챗봇의 도메인 범위 결정

* 1. 학사일정 : 종강/개강, 공모전, 수강신청
  2. 학과정보 : 교수진, 전공과목, 교양과목, 학과 정보
  3. 건물정보 : 부속건물, 식당

위와 같은 정보들을 제공해주는 챗봇을 완성 후 차후 업데이트

해당 정적인 정보들의 학습이 끝나면 FLASK를 이용해 카카오톡 API와 연동하여 최종적으로 카카오톡 플러스 친구를 형성.

해당 챗봇과 대화를 진행하며 챗봇을 업데이트 및 보완.

정보 제공의 경우 정적인 답변을 하므로 어려움이 없지만

유저와 실시간 소통이 필요한 동적인 대화(ex, 잡담, 분위기 파악 등)을 구현하는 것에 대한 공부 필요.

## 현재 진행중인 Intent

Intent 형성 완료 : 날씨, 인사말, 부속건물 위치 및 정보, 대학설명, 학과 정보, 식당

Intent 진행 예정 : 수강신청, 종강/개강, 교양필수

자세한 코딩과 내용은 블로그에 정리하여 업로드 중입니다

블로그 주소 : <https://blog.naver.com/hts980521/223063792581>

7주차 : 2023.04.04– 2023.04.10

## 1. 부속건물 위치 및 정보 연동

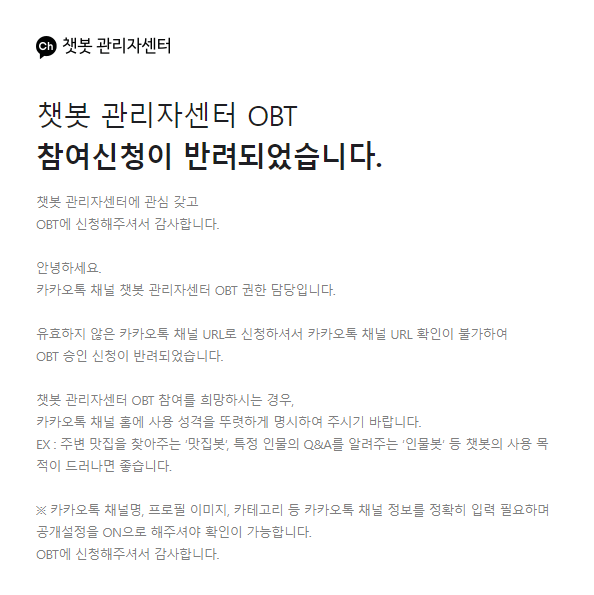
부속건물 위치와 정보를 json 파일로 만든 후 필요할때 불러와 답변하는 기능을

flask 서버에서 쓸 수 있도록 약간 수정

블로그 주소 : <https://blog.naver.com/hts980521/223064613492>

## 2. 카카오톡에서 서비스를 위한 허가 얻기

바뀐 정책에 따라 부분 유료화로 카카오톡 챗봇 서비스가 바뀜에 따라 카카오톡 오픈 채팅을 통해 챗봇을 서비스 하기 위해 허가증 요청을 하였으나, 공개설정을 open으로 하지 않은체 허가요청을 하여 일주일 밀리게 되었음.



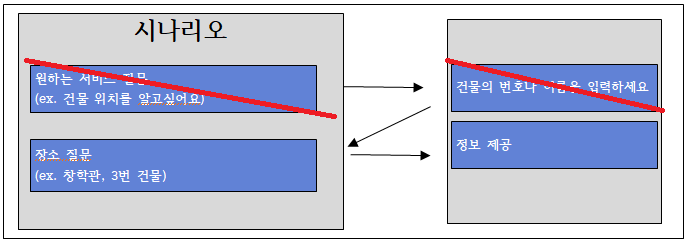
자세한 코딩과 내용은 블로그에 정리하여 업로드 중입니다

블로그 주소 : <https://blog.naver.com/hts980521/223063792581>

8주차 : 2023.04.11– 2023.04.17

## 부속건물 위치 및 정보 연동(수정)

캠퍼스안내 – 위치안내 & 부대시설 학교 홈페이지 캠퍼스 지도 참고해 json파일 업데이트

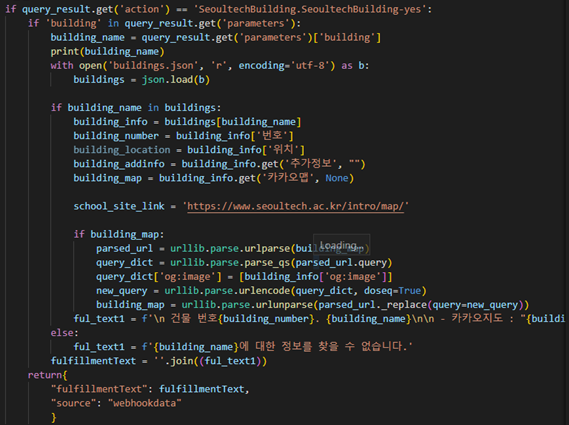


계획한 시나리오는 위와 같았지만 주변 친구들에게 정보를 제공하지 않고 위치를 찾는 질문을 유도한 결과 1번 시나리오는 필요가 없어 생략.

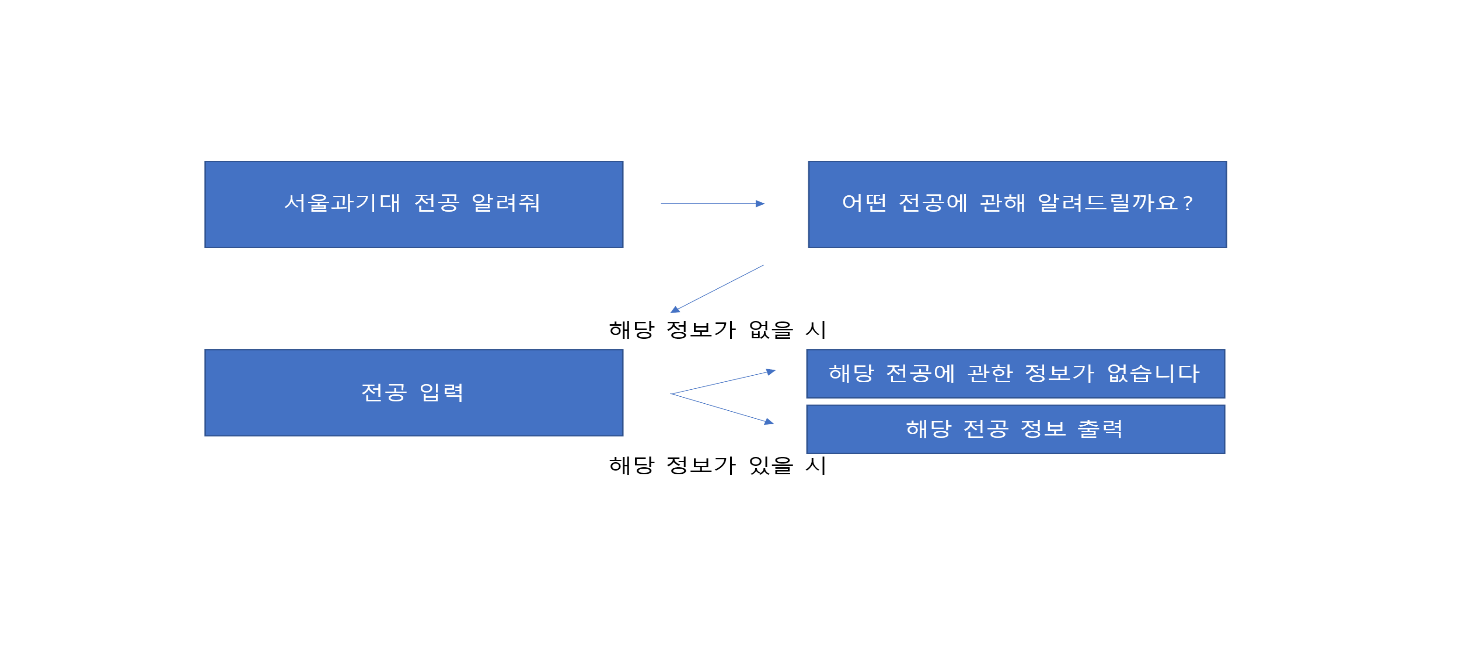


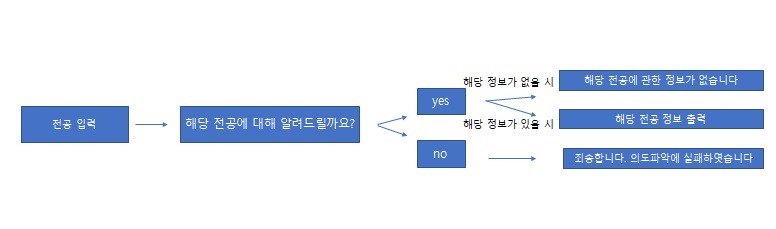
챗미를 기반으로 json파일에 각 건물마다 ● 건물이름, ● 건물번호, ● 위치링크(카카오맵), ● 추가정보(부대시설, 실사용 정보), ● 캠퍼스지도 링크 제공 + ● 카카오톡 플러스 친구로 연동할 계획이라 링크 썸네일을 건물 이미지로 변경하는 각각 이미지파일 저장.

엔티티 or json파일 내 정보를 통해 건물을 검색가능 (ex. 창학관 = 3번건물)



## 전공 정보



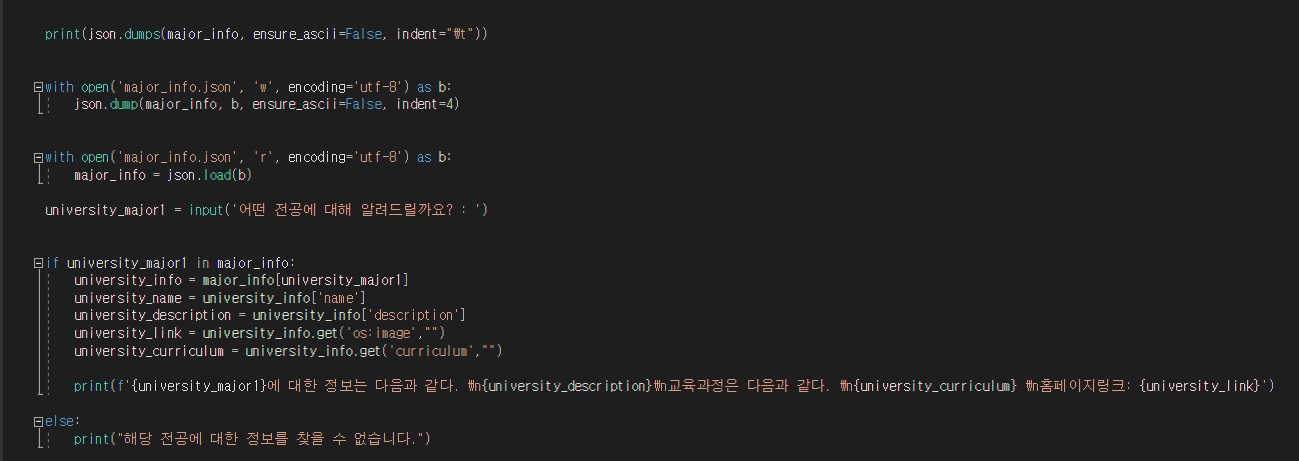
기존 follow-up intent에서 하위 intent(전공 바로 입력 시)로 바로 진입이 불가능하여 위 구조를 아래와 같이 수정

서울과기대의 단과대학(공과대학, 정보통신대학, 에너지바이오대학 등)에 속한 전공에 관한 정보, 커리큘렴, 각 학과의 홈페이지를 전달하는 기능

json파일 형태로 각 전공들에 대한 정보를 테이블로 정리(세부 내용은 블로그에 정리)

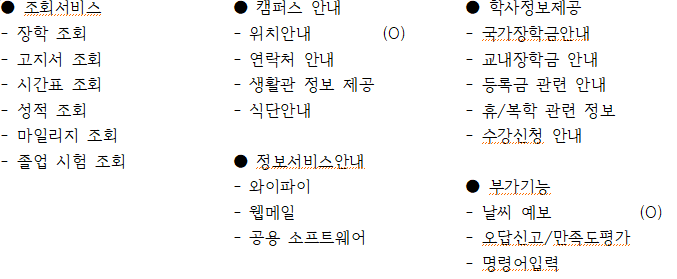
dialogflow entity에서 각 전공에 관한 동의어 정의

Intent에 따라 사용자가 입력한 전공에 관한 정보 전달



## 3. 챗봇 데이터 구축 방향성

챗미 가이드라인 - <file:///C:/Users/th/Downloads/chatbot_manual.pdf>



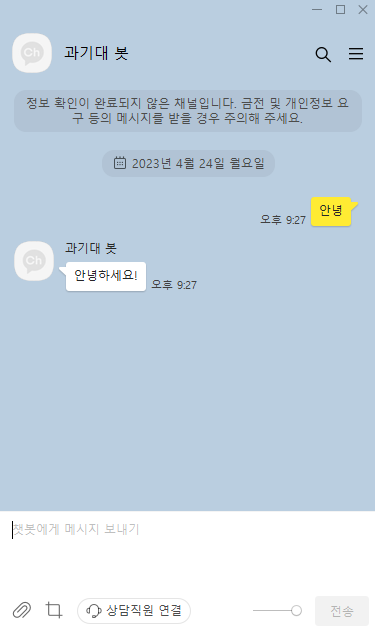
크롤링, 데이터베이스 JSON파일, API 3가지 방법으로 챗봇의 기본 데이터를 구축하는 법을 공부하여 실행까지 마침

챗미를 참고하여 추가적인 데이터를 제공하는 챗봇이 목표이기에 가이드라인에 올라온 21가지 항목중 조회서비스를 제공한 나머지 항목을 각각 맞는 방법으로 업데이트 하고 카카오톡 플러스 친구에 연동하는 것이 목표.

9주차 : 2023.04.18– 2023.04.24

## 1. 카카오톡 채널과 연동

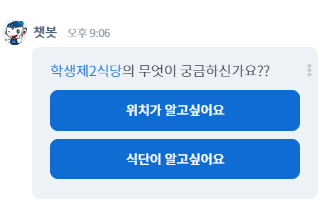
카카오톡 채널을 만든후 오픈빌더 발급 이 완료되어 지금까지 구축한 챗봇을 카카오톡과 연동 하였다.



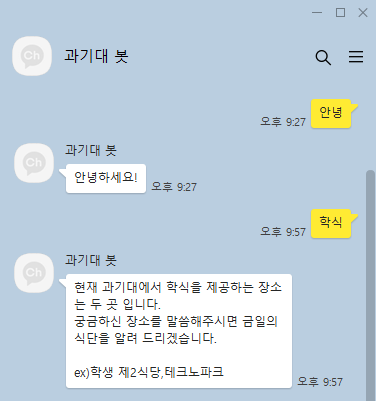
코드 : <https://blog.naver.com/PostView.naver?blogId=hts980521&Redirect=View&logNo=223084023517&categoryNo=12&isAfterWrite=true&isMrblogPost=false&isHappyBeanLeverage=true&contentLength=1665&isWeeklyDiaryPopupEnabled=false>

## 2. 방향성 고민

기존의 만들어진 '챗미' 의 경우 건물의 이름을 질문 하였을 경우 건물의 무슨 정보를 묻고 싶은지의 정보를 타이핑이 아닌 버튼식으로 해결한다.



서비스 하는 플랫폼이 카카오톡이 되어버린 이상 버튼식 질문을 안 할 이유가 없었고, 기존의 dialogflow에서 제공하는 chatbot의 시나리오 부분과 자연어 처리 부분을 카카오톡에서도 제공하는것을 확인 하였기에 dialogflow의 사용이 애매해져 버렸다.

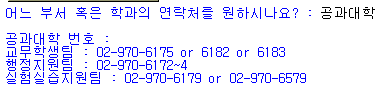


10주차 : 2023.04.25– 2023.05.01

## Dialogflow Intent 추가

* 1. 연락처 (부서, 단과대, 식당, 도서관 등)

Json 형태로 연락처를 정리하여 사용자 발화가 연락처 intent로 들어갔을 때 필수 파라미터인 부서명 혹은 학과명이 없다면 이를 물어보고, 해당 파라미터 정보에 따라 연락처 제공



* 1. 휴/복학 정보



Dialogflow와 마찬가지로 10개 이상의 엔티티를 활용한 사용자 예상 발화를 입력하고 자연어 처리를 통해 “휴학” or “복학”과 관련한 내용이 나오면 연동이 되도록 구현했다.

Dialogflow와 다르게 사용자의 질문방향이 정해지지 않은 반면 카카오톡의 같은 시나리오안에 블록을 구성하여 버튼으로 각각의 블록으로 연결할수 있고 사용자의 질문 방향도 유도할수 있다.



간단하게 실행한 화면이다. 카카오톡에 제공되는 추가적인 서비스를 활용하여 질문의 주제와 다른 시나리오지만 관련 되있는 (ex. 휴학 <-> 복학) 주제를 연결 지을 수 있는 버튼을 구현했다.

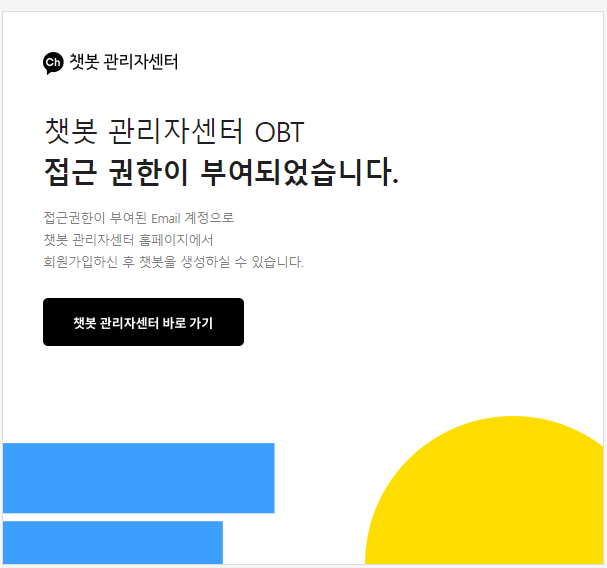
* 1. 식단 정보

학식정보를 물어보았을 때, 학식을 제공하는 5곳의 건물 중 알고 싶은 건물을 물은 후 홈페이지에서 크롤링한 데이터를 가져와 정보 제공

<https://blog.naver.com/PostView.naver?blogId=hts980521&Redirect=View&logNo=223090744448&categoryNo=39&isAfterWrite=true&isMrblogPost=false&isHappyBeanLeverage=true&contentLength=23002&isWeeklyDiaryPopupEnabled=false>

## 카카오톡 챗봇 채널 연동 및 학습

2-1. 새로운 카카오톡 아이디에서 챗봇 연동 완료



2-2. 카카오 챗봇 용어 정리

1. 엔티티 : Dialogflow와 동일하나 CSV, xlsx 파일 업로드 가능 (단, 파일은utf-8로 인코딩),

기본적으로 제공되는 엔티티는 블로그에 정리

1. 시나리오 : 봇 안에서 사용자가 경험할 수 있는 서비스 단위.

예를 들어, 서울과기대의 정보를 제공하는 챗봇의 경우 이 챗봇이 제공할 수 있는 서비스 단위, 즉 시나리오는 ‘식단’, ‘전공’, ‘휴/복학’ 등이 될 수 있다.

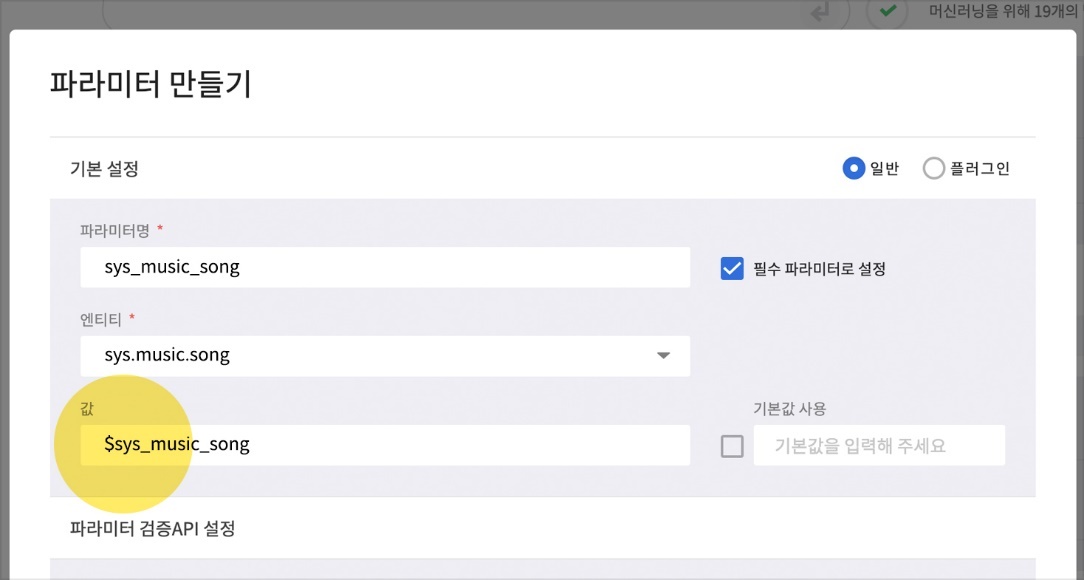
기본적으로 제공되는 시나리오는 Welcome, Fall-back, Exit블록을 포함한다.

1. 블록 : Dialogflow에서의 intent와 유사
2. 스킬 : Dialogflow에서의 fulfillment와 유사
3. NLU : BERT Language 모델 사용. Dialogflow CX 버전과 동일한 NLU

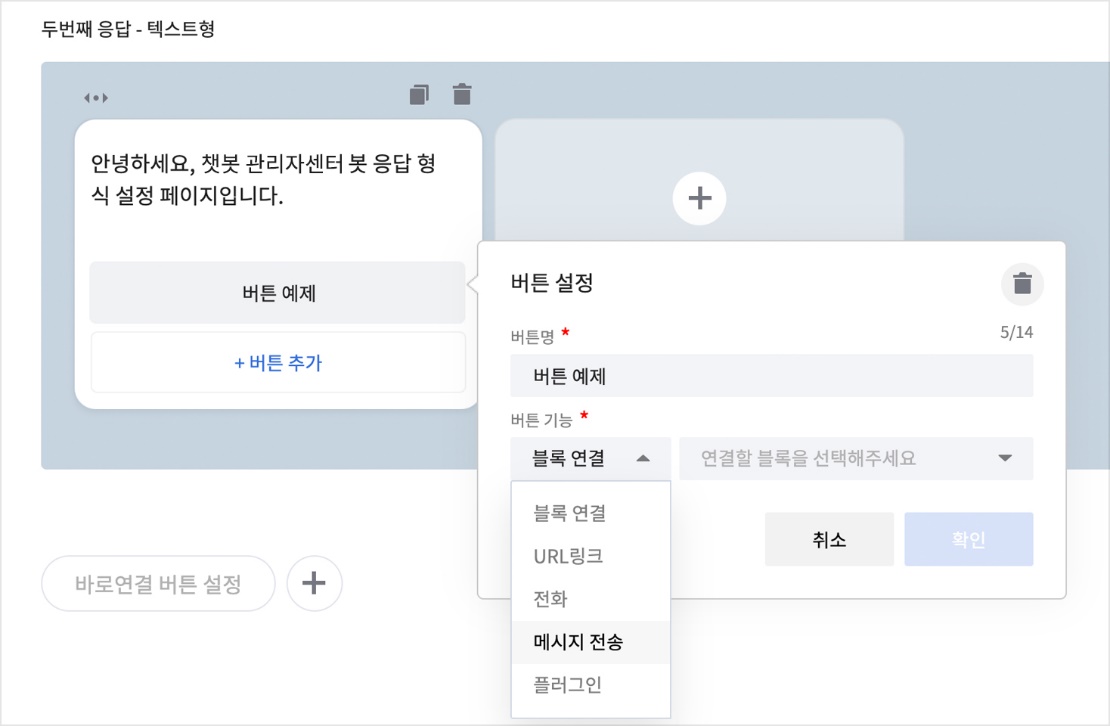
더 자세한 내용은 블로그에 공부하며 정리 : https://blog.naver.com/hts980521/223090635267

2-3. Dialogflow와 카카오톡 챗봇과의 차이점 :

1. 파라미터 값 : 엔티티뿐 아니라 사용자의 발화나 고정된 값을 파라미터 값으로 설정 가능



1. 응답 유형 : 텍스트뿐 아니라 버튼, 이미지, 카드 , 리스트 등 여러 유형의 답변기능 제공



블록(intent)끼리의 연동도 dialogflow와 달리 양방향으로 가능.

1. 스킬 : 언어로는nodejs 웹프레임워크로는 expressjs 권장. 하지만 기존의 flask를 사용한 상태로 연동가능한지 공부 중.

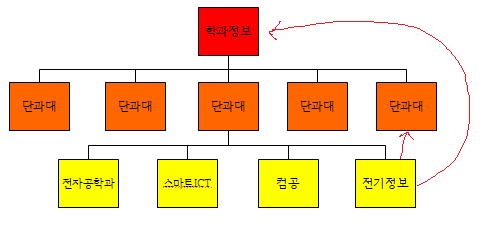
## 차후 목표

1. 스킬, 플러그인, 블록 연결 등 기존의 dialogflow에 없던 기능들에 대한 공부

2. Dialogflow와 카카오톡 챗봇의 기능 및 성능 비교

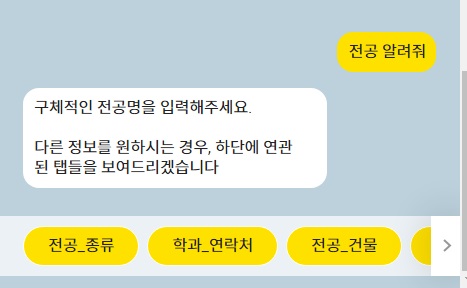
11주차 : 2023.05.02– 2023.05.08

## Block 정리

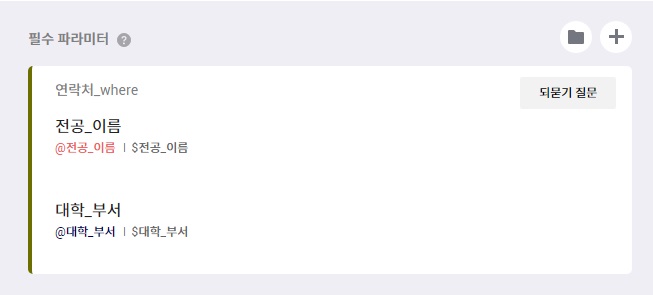


기존 Dialogflow에 입력했었던 날씨, 건물, 학과정보 등 카카오톡에 맞는 방식으로 옮김

dialogflow에서 구현했던 방식과 달리 각각의 블록을 형성할수 있으며 최종 블록에서만 서버에 구축해 둔 데이터를 출력하는 방식으로 구현. 블록간의 상하이동, 또는 다른 시나리오로 이동 가능.



구체적 전공 이름이 없을 시 되묻기 질문 + 연관된 정보를 하단 탭에 제공.

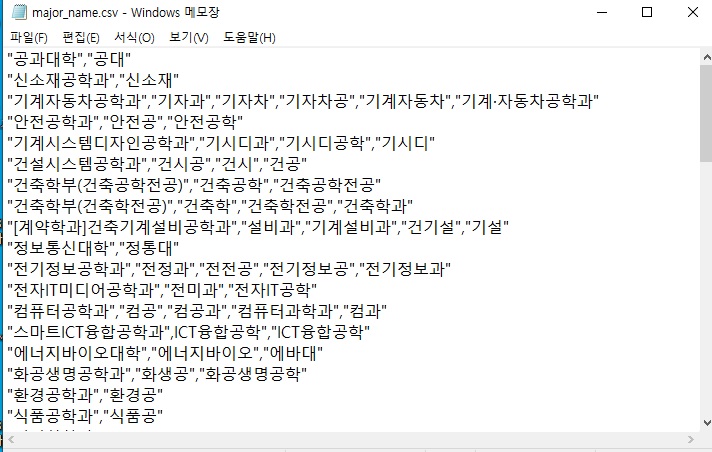


연락처 시나리오에서 필수 파리미터 두 개를 그룹으로 연결하여 entity가 섞이는 문제를 해결

엔티티 추가중

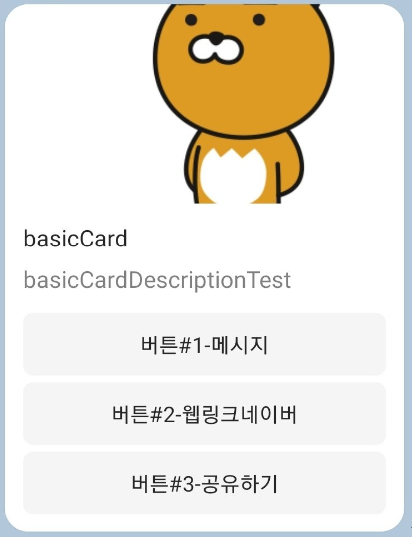


엔티티는 csv파일로 메모장(혹은 엑셀)에 정리하여 한번에 업로드

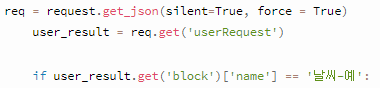


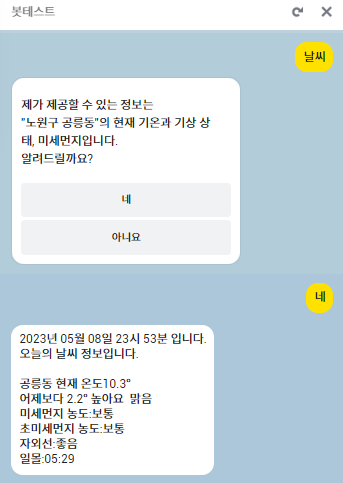
## 데이터 연동 & 버튼 동시 사용

10주차까지 기존에 구현해둔 데이터들을 카카오톡과 연동을 하면 카카오톡의 장점인 버튼을 통한 구현이 불가능하다고 알고있었음.



카카오톡 “Skill” 기능을 통해 카카오톡의 json형식을 맞추면 데이터베이스의 정보와 버튼을 동시에 사용가능.

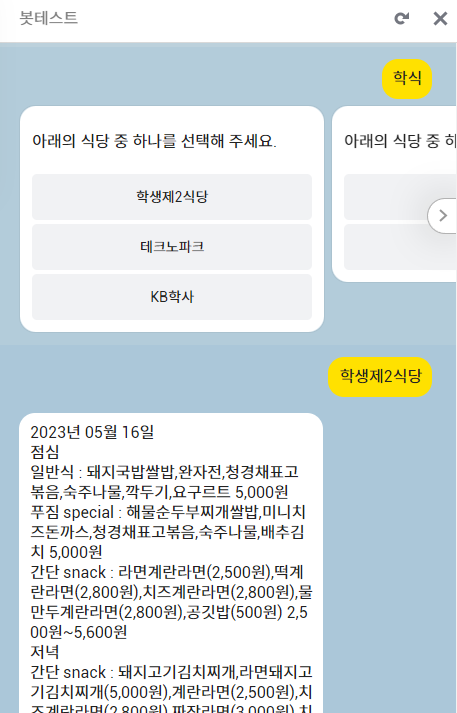
카카오톡에서 제공하는 가이드를 참고하여 스킬 실행시 카카오톡이 보내주는 json파일에 대한 정보 userRequest -> block -> name 탭으로 if문을 달아주면 하나의 URL로 여러 스킬을 달아 줄 수 있을 것 같아 수정했다.

텍스트 출력만 요구하는 날씨를 연동하였습니다.

이미지와, url, 버튼 출력을 요하는 다른 것들은 공부중

12주차 : 2023.05.09– 2023.05.15

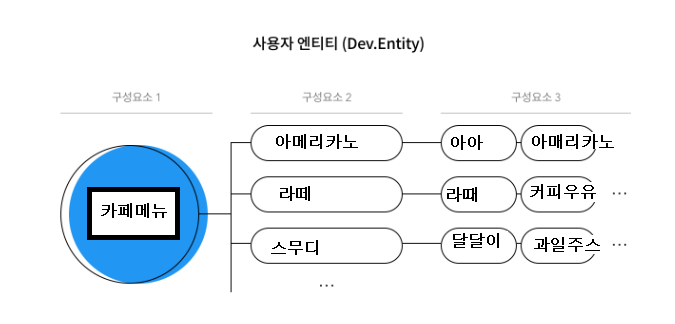
1. dialogflow 기능 옮겨오기



dialogflow에서 구현했던 학식 , 건물위치 시나리오를 카카오에서도 구현가능 하도록 하였다.

<https://blog.naver.com/hts980521/223103165367>

2. 새로운 시나리오 추가중 문제 발견, 대체방안 찾기



기존의 시나리오가 아닌 순서가 바뀐 질문이 나왔을시 응답이 가능토록 짜주는 과정에서 카카오챗봇은 dialogflow 와 달리 사용자 발화에 엔티티가 들어갔을시 추출값을 대표 엔티티 값으로 반환 못해주는것을 확인하였고 flask에도 똑같이 엔티티를 추가하는 것으로 해결.

<https://blog.naver.com/hts980521/223103165367>

3. 현재까지 완성도

● 캠퍼스 안내 ● 정보서비스안내

- 위치안내 (o) - 와이파이 (△)

- 연락처 안내 (x) - 웹메일 (△)

- 생활관 정보 제공 - 공용 소프트웨어 (o)

- 식단 안내 (o)

● 학사정보제공 ● 부가기능

- 국가장학금안내(ㅇ) - 날씨 예보 (o)

- 교내 장학금 안내 (ㅇ) - 오답신고 (o)

- 등록금 관련 안내 (ㅇ) - 만족도평가

- 휴/복학 관련 정보 (ㅇ) - 명령어 입력

- 수강신청 안내 (ㅇ)

- 전공 정보 제공 (ㅇ)

방학 중간 정검1 : 2023.05.16– 2023.06.25

1. **카카오톡 챗봇 엔티티 문제 업데이트**

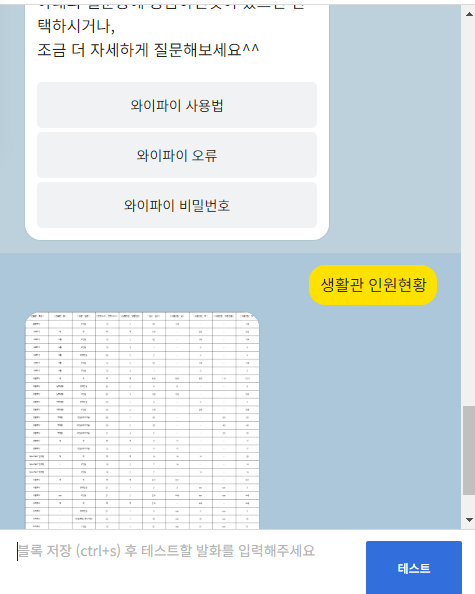


한달동안 카카오 챗봇 내부에서 업데이트를 하였는지 정상적으로 대표 엔티티를 출력해주기 시작하여. flask 내부에서 동의어 엔티티를 받을시 대표엔티티 값으로 변경해주던 코드 삭제

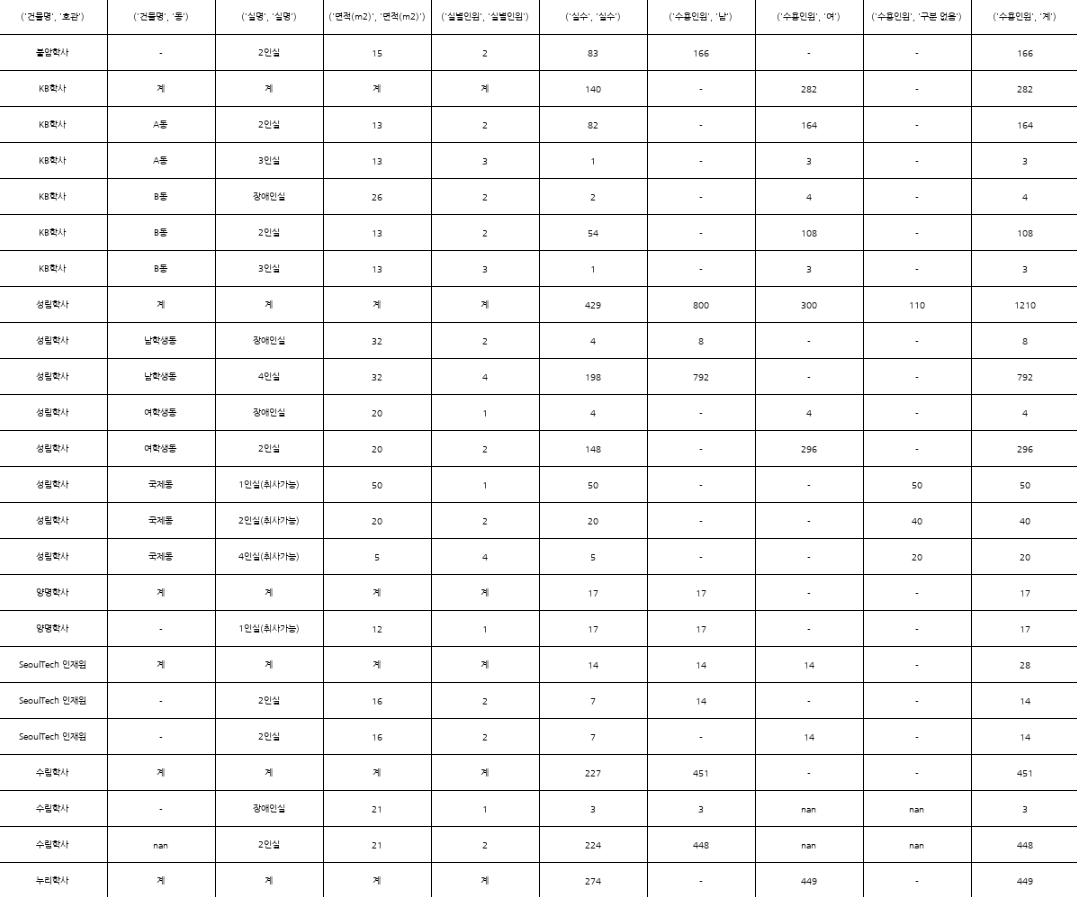
**2. 지속적인 발화 업데이트**

국가 장학금 안내사항 80프로까지 답변이 가능하도록 context 추가. (챗미와 비교하여)

1. **생활관 정보(크롤링 데이터 정리)**



생활관 관련 정보를 제공할 때 표를 제공해야 하는 경우가 많은데 해당 표의 rowspan과 colspan에 따라 적절하게 셀을 병합하고 표의 형식을 고려하는 부분에서 난항을 겪고 있다.



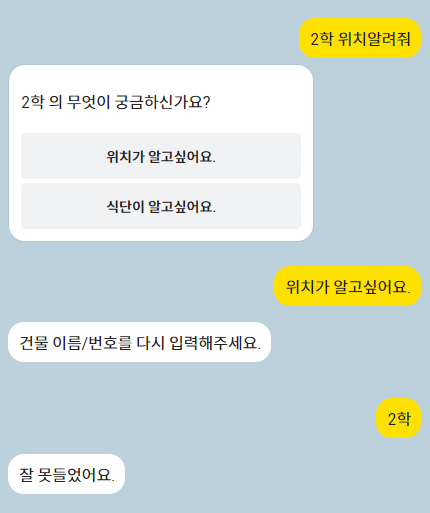
(rowspan 존재하는 항목들의 셀이 병합되지 않아 표가 깔끔하지 못한 상황)

1. **연락처**

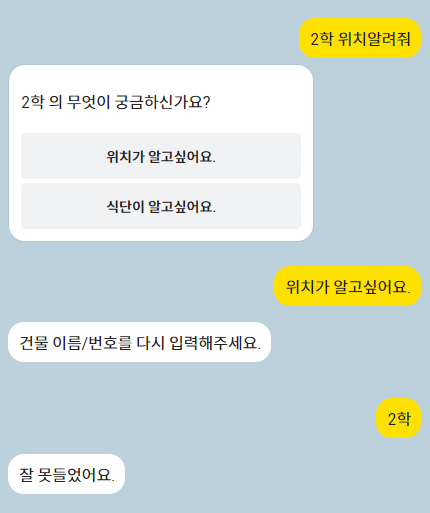
카카오 챗봇의 엔티티 문제가 해결됨에 따라 각 학부 및 전공 별 연락처 연동 중(json파일에 연락처를 정리하여 스킬과 연동, 현재 json 파일 작성 중)

방학 중간 정검2 : 2023.06.26– 2023.07.23

**1. 챗봇 오류 수정**



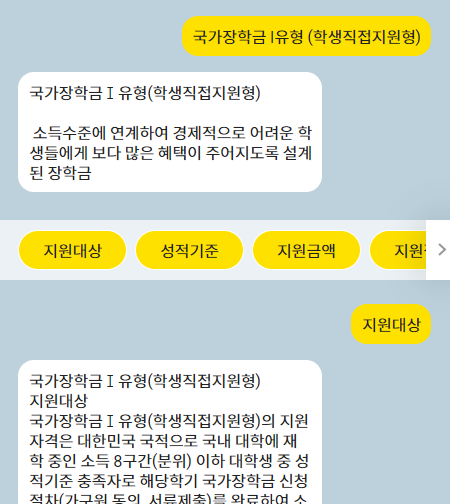
학식을 제공해주는 건물 조회시 사용자의 의도가 건물의위치를 묻는것인지 학식정보를 묻는 것인지 묻는 context를 추가 하였는데 건물 위치 버튼 클릭시 위치 조회가 되지 않는 오류를 발견하였다.



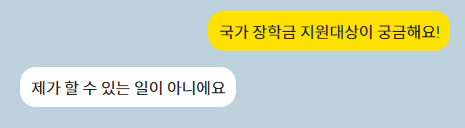
건물 entity를 나누는 과정에서 일반 건물과는 달리 위치가 아닌 다른 정보를 얻어야하는 건물들을 나누어 정리하였는데 이 과정에서 코딩에 나눈 건물들의 entity를 추가하지 않아 발생한 것으로 확인하였다.



2**. 챗봇 기능 고민**

****

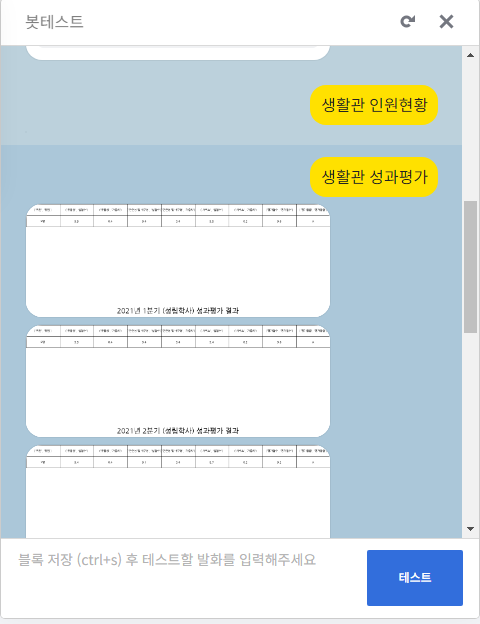
**그림3. 기존의 답변 방식 ( 총 6번의 질문에 대답하여야 답변 도출 )**

****

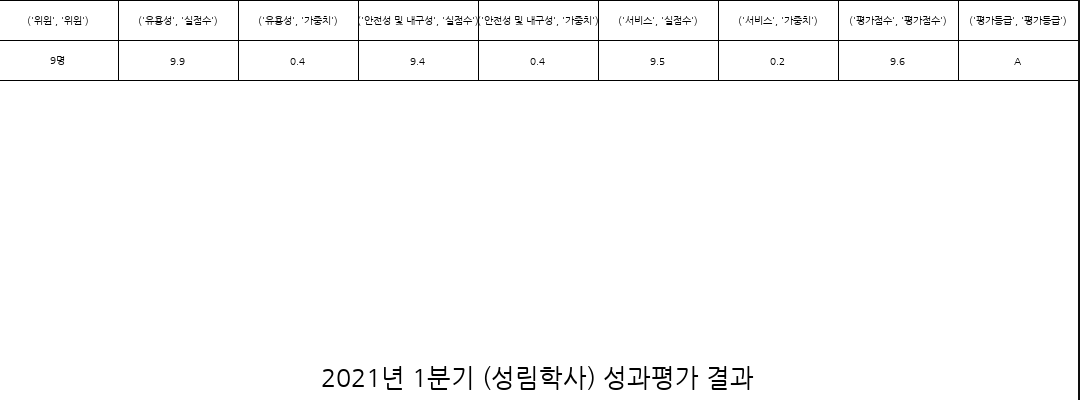
**그림4. 사용자의 질문에 따라 단계 skip의 필요성**

장학 조회시 사용자의 의도 파악이 중요한 만큼, '장학' 이라는 keyword 만을 받아 여러번 질문하여 사용자의 의도를 파악하는 방식을 채택하였지만, 굉장히 많은 답변 과정이 번거럽고 취지와 맞지않는다고 생각하여 사용자의 첫 물음에서 질문 단계를 skip할 방법을 생각중이다.

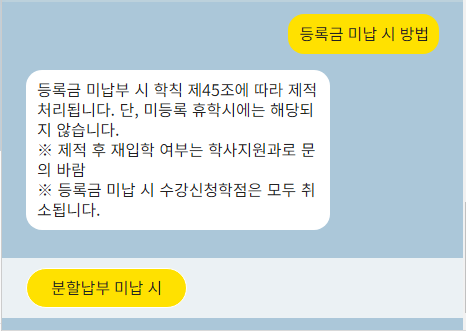
3**. 생활관 / 등록금 / 수강신청 시나리오 추가**



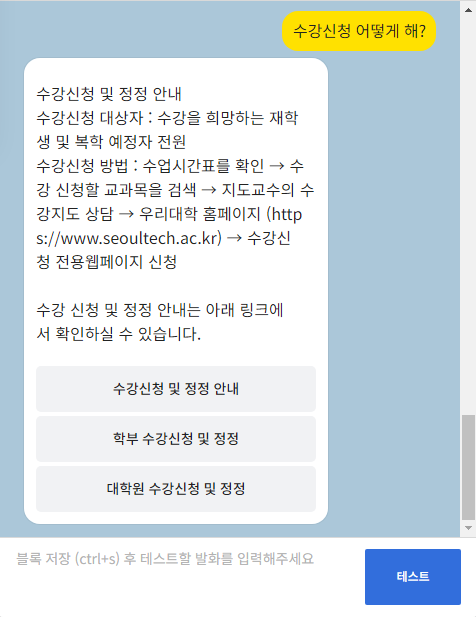
각 생활관(성림/누리/수림)에 따른 성과평가 결과 표를 크롤링하여 표 형태로 이미지를 출력해주도록 하였으나, 이미지의 크기와 디자인을 어떻게 해야 더 깔끔하게 구현할 수 있을지 고민중



표와 이름 사이의 여백을 지우고 테이블의 헤더를 더 자세하게 분류하도록 코딩 중



등록금의 경우 미납 / 납부 / 증명서 / 외국인 등록 등의 경우를 나누어 등록금 시나리오 작성



수강신청은 링크를 제공하여 수강신청 페이지로 유도하고 해당 과정을 상세하게 설명

(단, 챗미에서 제공하는 개인의 수강신청 현황은 교내 서버에 접속 불가하여 제공할 수 없음)

4**. 남은 일정**

대략적인 시나리오의 구성이 끝나, 챗봇의 테스트 및 현재까지 완성한 시나리오들의 오류와 발화목록을 체크할 필요가 있다.

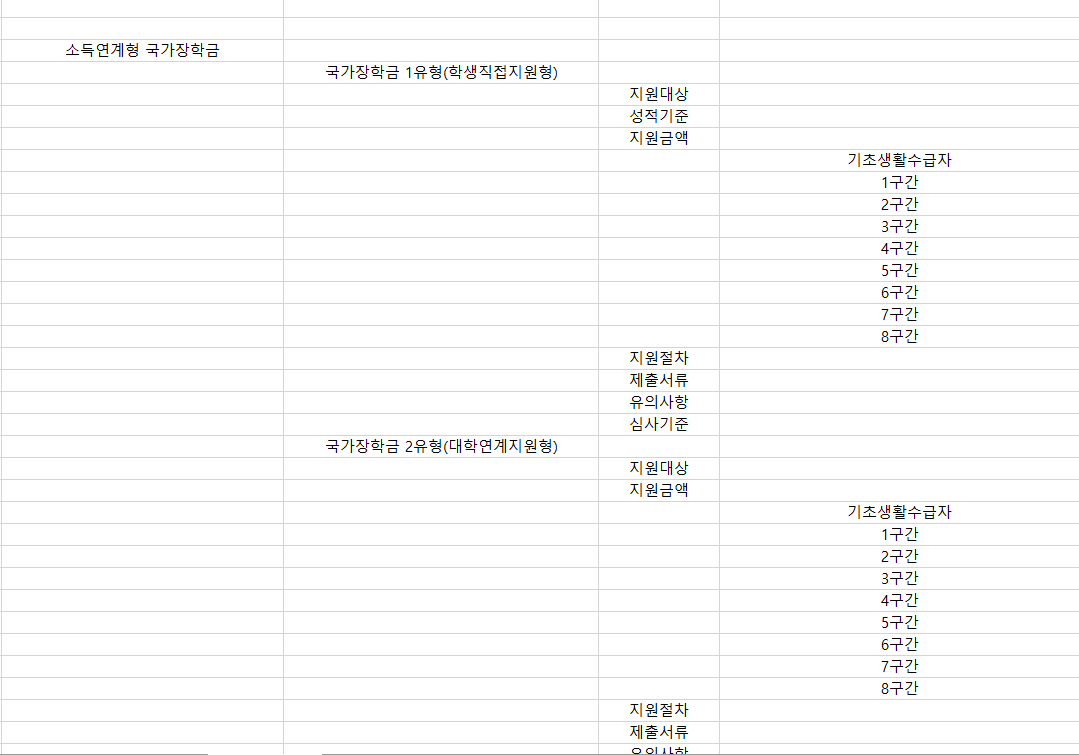
방학 전 목표로 했던 시나리오 외 신입생들에게 도움이 될 수 있는 시나리오들을 새롭게 구성할 예정이다.

2023.09.01 – 2023.09.09

1. **현재까지 진행한 시나리오 및 블록 정리**

**챗봇을 만들기 위해 참고하였던 챗미의 카테고리를 엑셀로 정리하여 지금까지 진행한 시나리오 및 블록들과 비교하였다.**

**(챗미 카테고리 :** <https://blog.naver.com/hts980521/223206898983>**)**

****

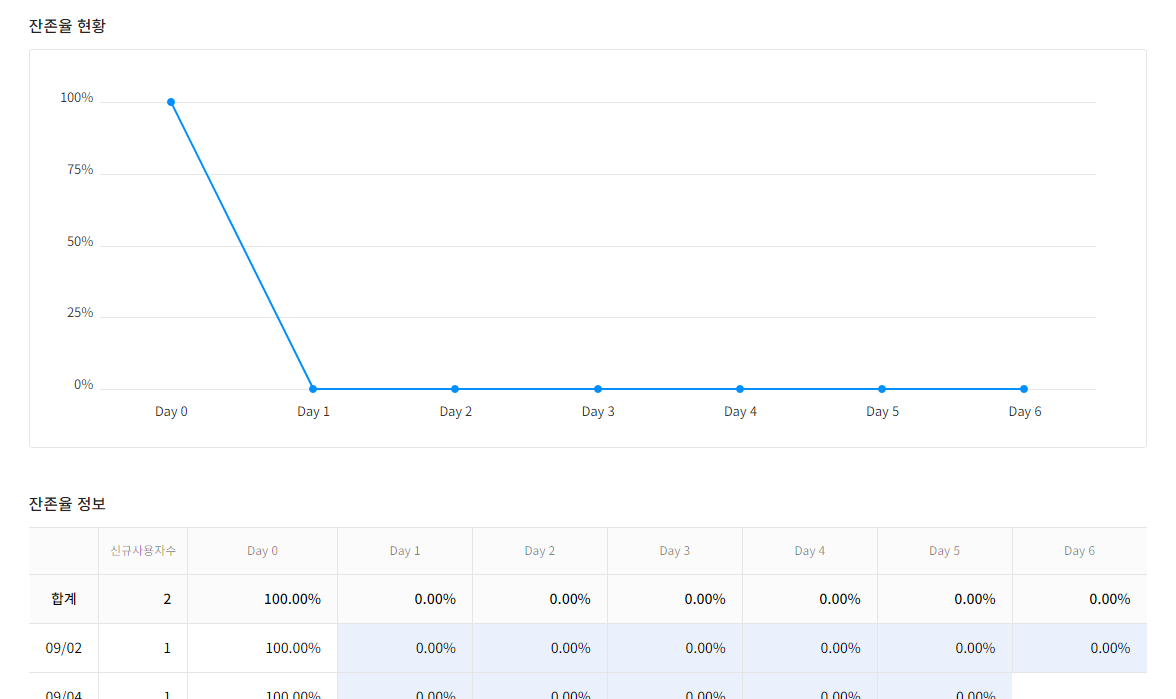
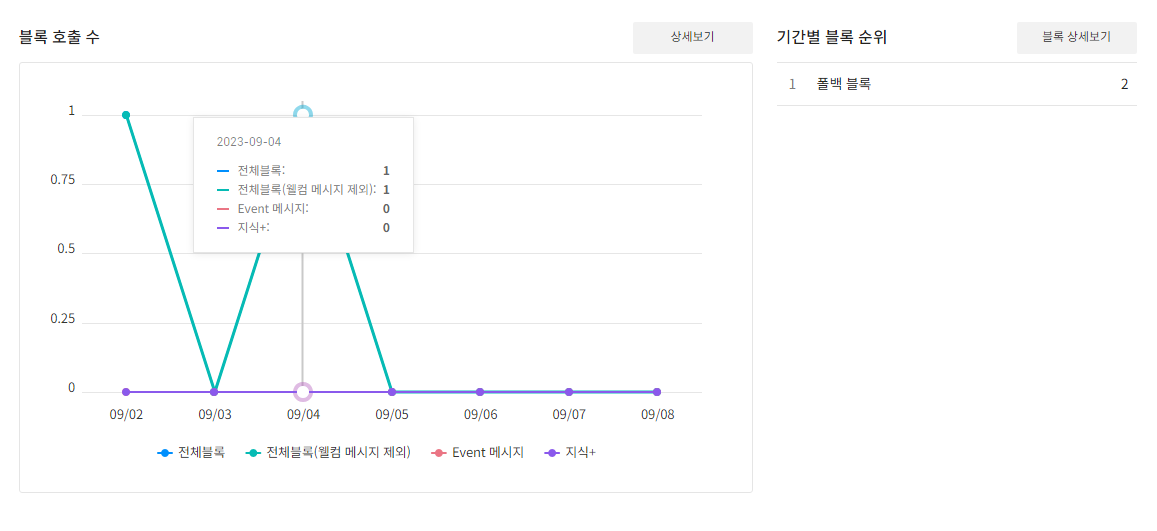
**원하는 질문의 블록으로 직접적으로 들어가는 것이 어려워 상위 질문부터 하위 질문으로 들어가는 챗미와 같은 방식으로 적용. \*구현 블록 수 : 약 800개**

1. **데모 테스트 예정**

커뮤니티와 지인들에게 데모를 배포하여 폴백 블록으로 들어가는 오류들을 정정 및 개선

카카오챗봇 관리자센터에서 사용자의 질문이 어떤 블록으로 들어갔는지 확인 가능. 위 기능을 통해 잘못된 블록으로 이동될 시, 알맞은 블록으로 이동되도록 수정 필요

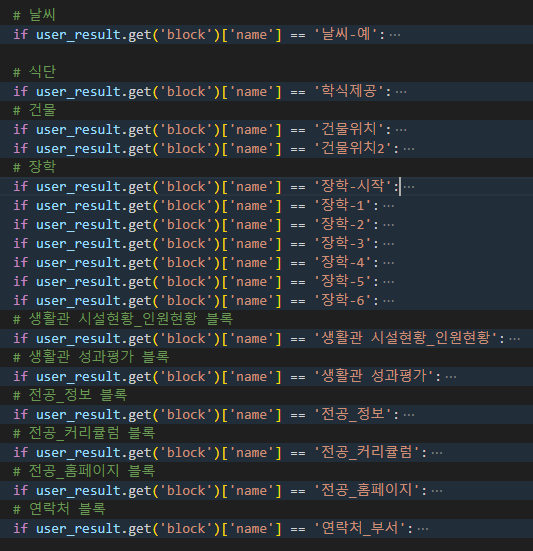
약 500개의 질문을 받아 응답률 개선.



2023.09.12 – 2023.09.18

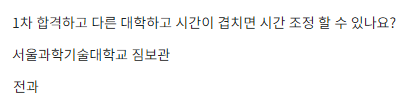
1. 코드 통합

서로 분할되어 있던 블록 코드들을 하나의 코드로 통합하여 서버의 활성화 / 비활성화를 편리하게 하도록 함.



1. 챗미와의 차별점

카카오톡 채널을 사용하여 챗봇을 배포하기 때문에 사용자 접근성이 편이하고 빠른 피드백 가능

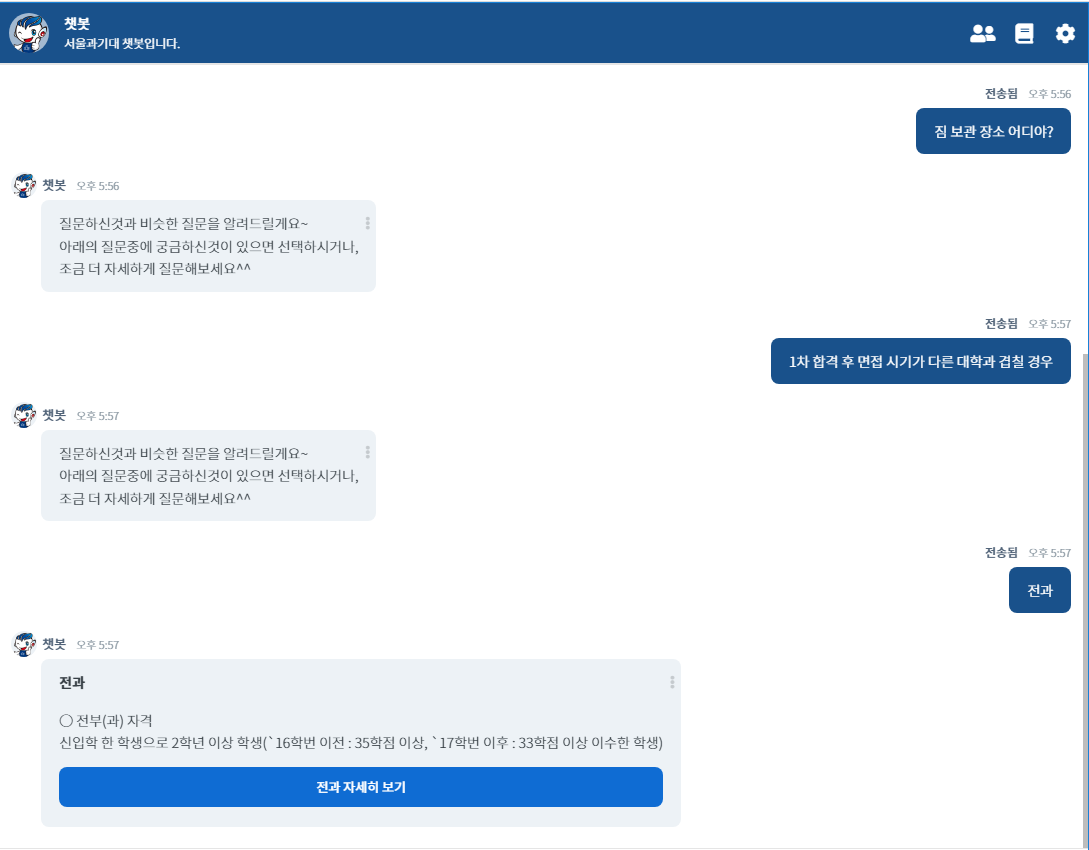


(방학 기간 중 카카오톡에서 실제 챗봇을 사용한 사용자의 질문)

1) 짐 보관 장소(사물함) 및 분실물 보관 장소

2) 신입생이 수시 1차 합격 이후 면접일이 겹칠 경우

3) 전과 관련 정보

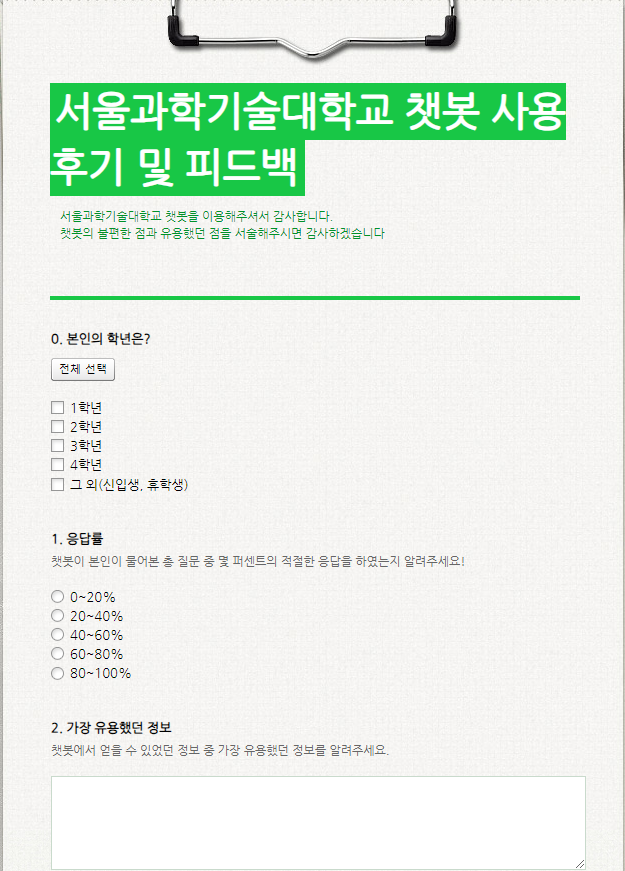


3가지 질문 중 1)과 2)는 챗미에서 제공하지 않는 정보. 실제 신입생, 재학생들이 필요한 정보를 블록으로 추가하여 챗봇 개선 가능.

1. 발화 매칭



사용자의 질문이 올바른 블록으로 들어가도록 학습되지 않은 발화들을 적절한 블록에 매칭시켜주는 기능을 통해 챗봇의 성능을 향상

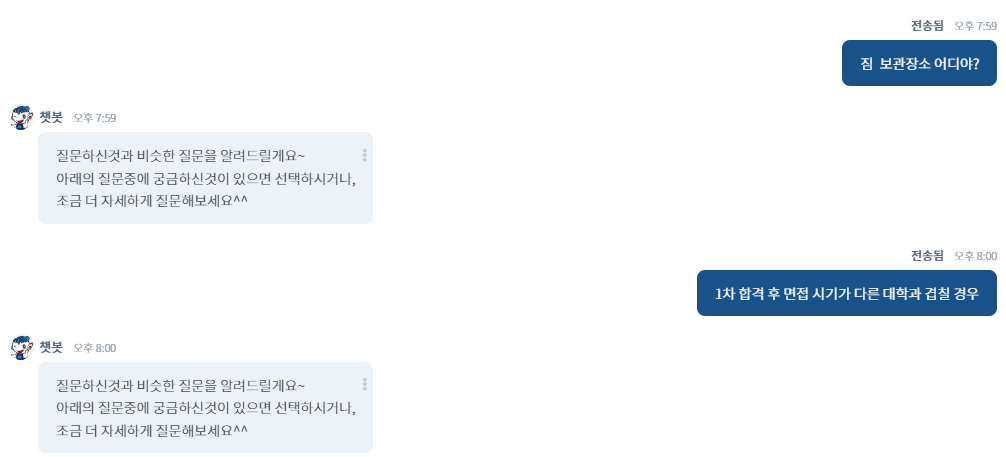
1. 데모 테스트 이후 설문조사

(네이버 폼 활용)

데모 테스트 이후 사용자에게 설문 조사를 통한 피드백 반영 예정.  
(카카오톡 설문링크 <https://naver.me/FkjTh8zG)>

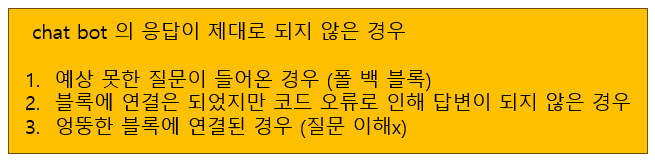
2023.09.12 – 2023.09.18

**1. 챗미 업데이트 확인**



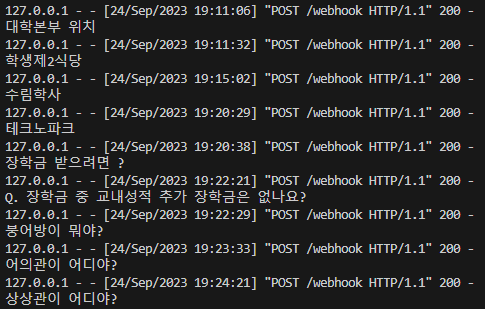
지난주에 질문하였던 질문 그대로 챗미에 물어보았지만 업데이트가 되지않았다.

**2. 사용자의 발화와 답변을 정확히 수집하기 위한 코드 작성**

****

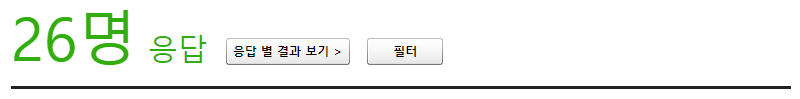
완성도 높은 피드백을 위해서 사용자의 발화와 그에따른 챗봇의 답변을 받기 위해서 코드 작성하려 했으나, dialogflow 와는 달리 챗봇의 접근 권한이 제한적이기에 스킬데이터가 활성화 되었을때만 ( 스킬로 답변된 블록이 호출되었을 경우) 사용자의 발화를 나오게 할 수 있었다.

또한 dialogflow 와는 달리 챗봇의 답변이 text 형식만 있는것이 아니기에 발화에 따른 답변이 무엇이 나왔는지도 확인하기 어려웠다.



**3. 챗봇 1차 데모 테스트 진행**

대략 30명으로 부터 최소 10개 이상의 질문을 받았고, 26명으로 부터 사용 후기 및 피드백을 받았다.

****

**4. 앞으로의 방향성**

후기를 대략적으로 살펴본 결과 챗봇의 업데이트 방향성을 잡을 수 있었다.

1. 정보제공 미흡

- 챗봇에 학습이 안된 경우

ex) 총장님 성함, 대학교 면적, 도서관 운영시간

ex) 과기대 근처 정보

- 챗봇이 학습하였으나 대답이 미흡하다 느낀경우

ex) 전공 필수과목 질문

2. 챗봇에 학습을 하였으나 답변이 안된경우

- skill을 사용하여 대답할 경우

ex)발화빈도를 1로 설정하여 같은 주제를 계속해서 물어볼시 이해못하는 현상

- 사용자 발화를 풍부하게 학습시키지 못한 경우

ex) 학과 홈페이지는 이해 못했지만 과기대 홈페이지는 이해

3. 챗봇의 잘못된 학습으로 오답을 내놓는 경우

- 숫자를 입력시 건물번호라 인식하여 건물위치를 답변함

- 연락처를 질문한적이 없는데 연락처를 답변함

2023.09.26 – 2023.10.05

**1. 항시 이용할수 있는 서버**

**A. 라즈베리파이를 활용하여 상시 서버 유지 (컴퓨터 상시 켜놓기 현실적 힘듬)**

**B. 네이버, 아마존 등 플랫폼에서 코드를 기간제로 실행하는 서버를 제공하는것도 있는 것 같다.**

**2. 48% 응답률 -> 70% 이상의 답변율 목표**

**A. 가이드라인을 제공한 주제에서는 응답률이 높고 활용도가 높았다. But 가이드라인이 없이 자유롭게 질문을 받은 경우 예상하지 못한 질문 + 가지고 있는 정보 활용이 낮음 -> 모든 주제에 대해서 제공할수 있는 데이터를 최대한으로 활용할수 있게 가이드라인을 제공하는 방식으로 변경할 예정**

**3. 지인 이외의 현실적인 사용방법**

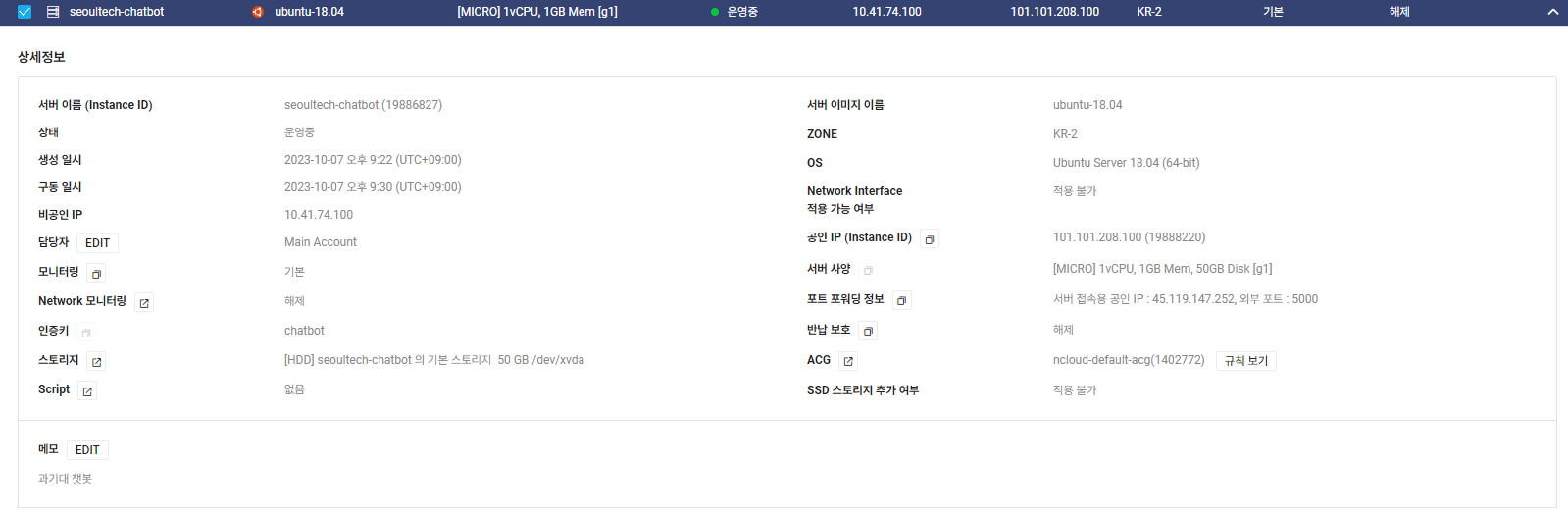
**4. 부족한 부족한 응답에 대한 피드백 +블록 추가 +스킬 오류수정**

**A. 답변이 나가지 못한 내용들에 대한 추가**

**B. 기능적 문제 (ex, 버튼을 여러 번 눌렀을 때도 오류가 발생하지 않도록 바꿈)**

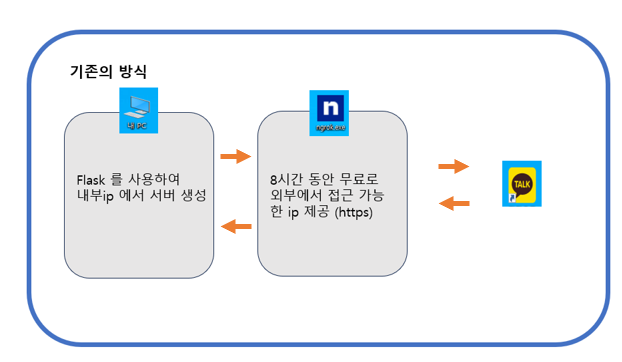
2023.10.06 – 2023.10.09

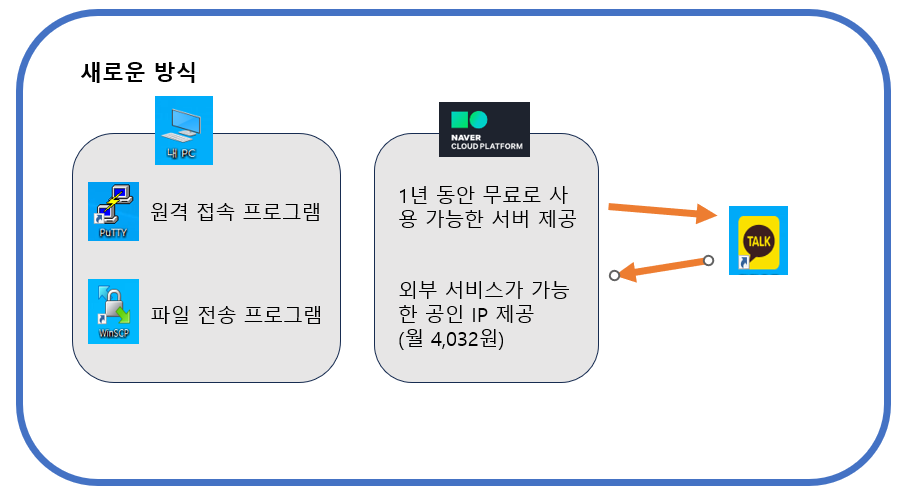
**1. 새로운 서버**

****

**기존의 방식으론 최대 8시간 유지가 한계였던 문제점을 고치고자 새로운 대안이 될 서버를 제공해주는 '네이버 클라우드 플랫폼' 서비스를 신청하여 1년간 무료로 사용가능한 서버를 만들었다.**

서버 생성 참고 : https://m.blog.naver.com/nieah914/221609709142





**장점**

**1.서버 상시 가동 \* (aws)**

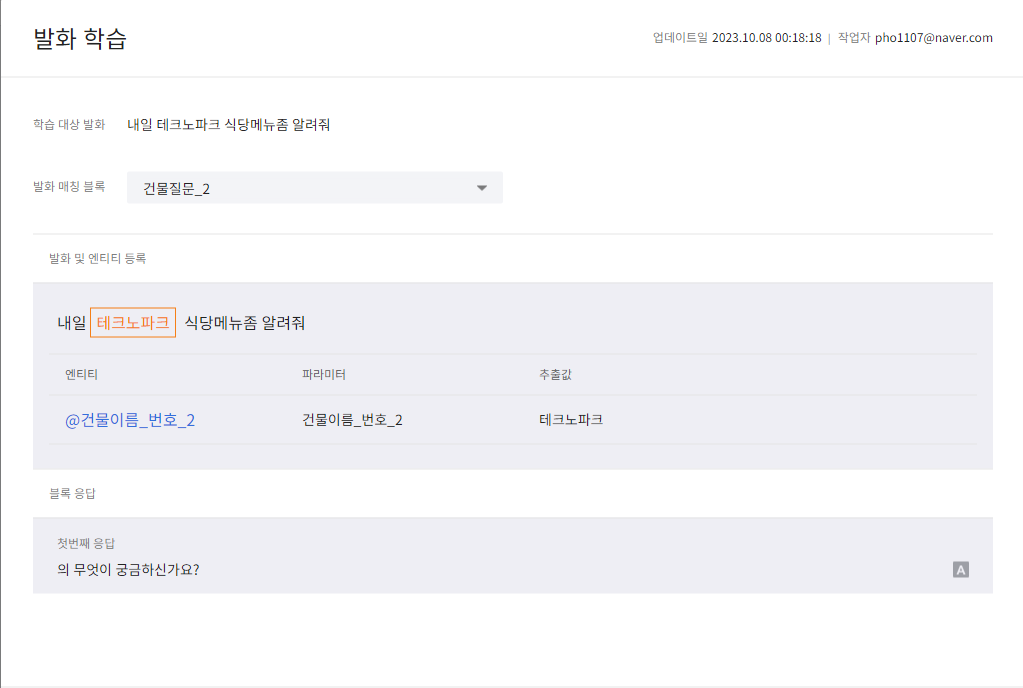
**단점**

**1.비용 발생**

**2.공인 IP (http) 로 생성되어 에러 발생하는 블록 발생 (ex. 생활관 인원현황)**

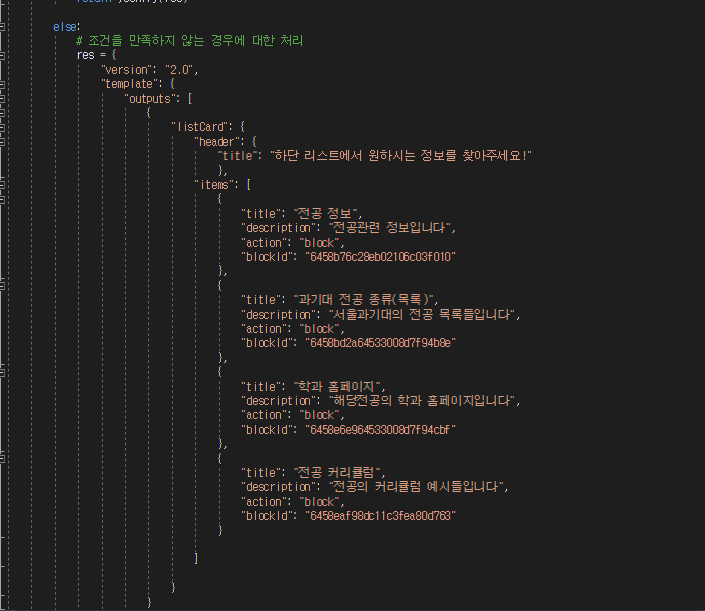
**2. 발화 매칭**

**해당 발화의 의도와 맞는 block이 존재하지만, 발화의 학습 불충분으로 인해 원하는 block으로 들어가지 않은 사용자 발화를 올바른 block으로 매칭시켜 학습.**

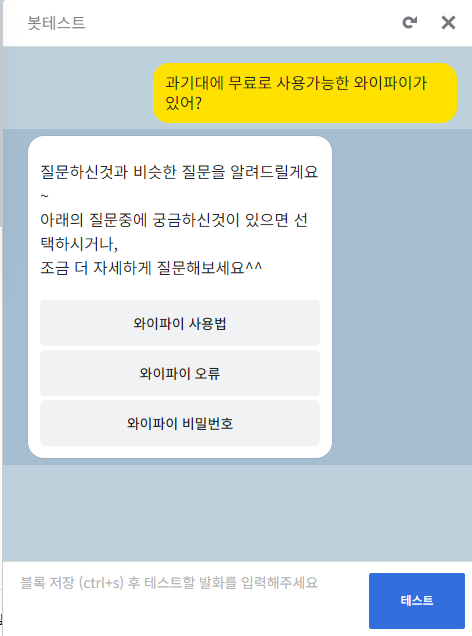
****

**3. 챗봇의 응답률 향상**

**3-1) 리스트 형태의 키워드 기반 선택지 제공**

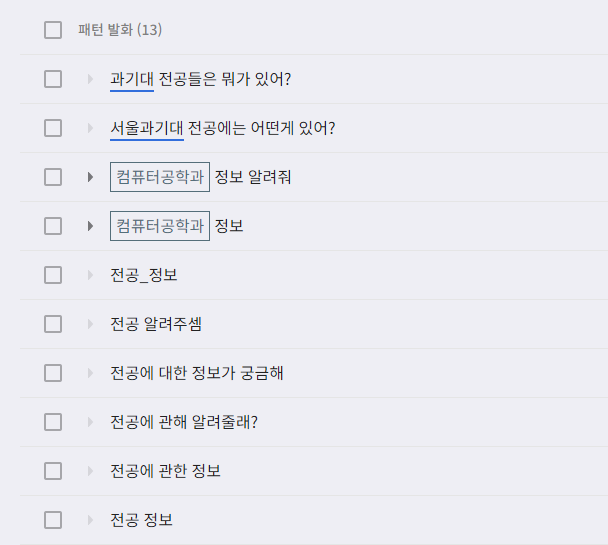
****

**기존에는 키워드는 입력 받았으나 학습하지 못한 발화가 입력되거나 의도한 block이 존재하지 않을 경우, “일치하는 정보를 찾을 수 없습니다” 라는 단순 text를 출력하였으나, 해당 키워드와 관련한 항목들을 list형태로 출력하여 사용자에게 선택지를 제공함으로써 응답률을 향상시키는 방법을 채택.**

****

**Ex) 키워드 “와이파이”와 관련된 질문이지만 해당 발화를 챗봇이 학습하지 못한 경우, 해당 키워드와 관련한 정보들을 리스트 형식으로 제공하여 사용자가 원하는 정보를 찾을 수 있도록 하여 응답률을 높인다.**

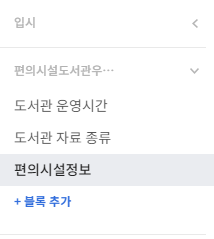
**3-2) 발화 학습**

****

**기존에 학습시킨 패턴 발화가 부족했던 경우 적정 개수(10~15개)에 맞게 추가하여 학습 진행**

**더 다양한 사용자의 발화에 대해 응답하도록 학습.**

**4. 새로운 응답 추가**

****

**데모 테스트를 진행하며 기존에 생각치 못했던 block들을 추가**

**5. 차후 일정**

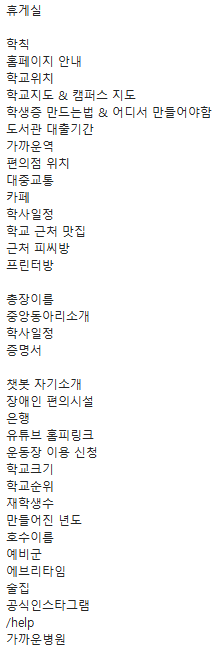
**발화 매칭, 코드 오류 수정, block의 추가 등을 통해 응답률이 실제로 향상되었는지 데모 테스트를 다시 진행하며**

**이전과의 응답률 비교 및 지속적인 챗봇 개선.**

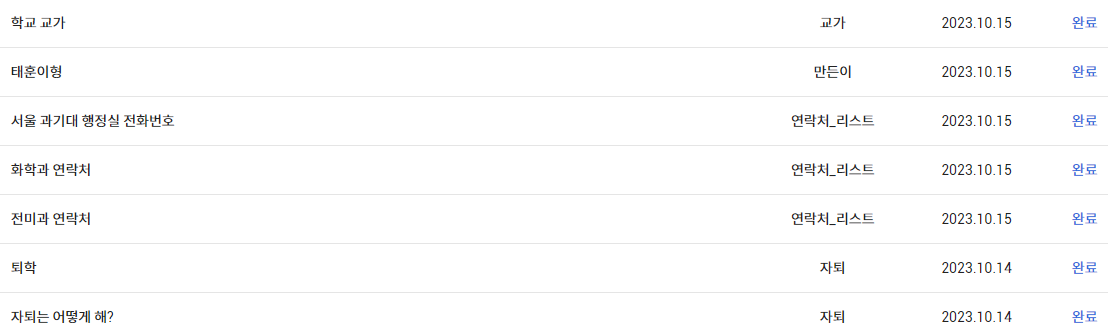
2023.10.10 – 2023.10.16

**1. 발화 추가**

**9월 14일날 진행했던 1차 데모테스트의 평가를 토대로 블록 추가 및 발화 학습을 진행하였다. (300개 이상의 발화 학습)**

****

**(그림1. 추가된 블록)**

****

**(그림2. 이미 있는 블록에 학습 발화 추가)**

**2. 오류 수정**

**카카오 챗봇 빌더에서 제공하는 오류알람기능을 참고하여 오류가 뜬 블록의 오류 수정을 진행하였다.**

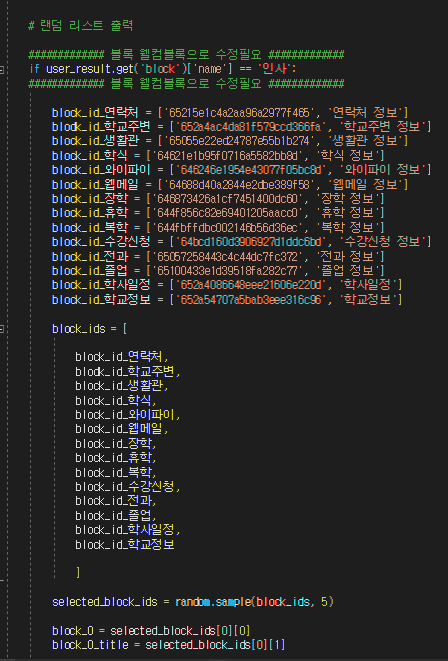
1. **블록과 코드의 충돌 수정**
2. **카카오 챗봇에서 구현한 블록과 코드가 충돌하여 블록에서 발생하던 오류 수정(in 전공 블록)**
3. **연락처 블록에서 다른 블록으로 넘어가지 않고 계속 연락처 블록에서 루프하는 오류 해결(코드 오류 수정), 연락처가 정상적으로 출력되지 않던 현상 해결**
4. **크롤링 중이던 홈페이지의 자료 변화로 인한 오답 전송**
5. **2학기 학식을 제공하지 않아 학식table이 없어져 오류가뜨던 KB학사 학식제공 코드 수정 (in 건물\_질문\_2 블록)**
6. **생활관 성과평가 홈페이지 업데이트로 인하여 잘못된 정보 제공중인 성과평가 블록 수정 (in 생활관\_성과평가 블록)**

****

**(그림3. 오류 발생 블록 목록)**

**3. 기능 추가**

1. **챗봇 활성화시 상위 블록 5개의 정보를 랜덤 추천하는 기능 추가**

****

**사용자에게 도움이 될 수 있는 상위 블록 5개를 랜덤하게 추출하여 사용자에게 추천 리스트를 보여주는 기능 추가**

**4. 차후 일정**

**10월 15일 부터 진행한 2차 데모테스트 자료 수집 후 새로 추가한 기능들이 챗봇 기능 향상에 얼마나 도움이 되었나 확인 및 지속적 업데이트 예정**