챗봇 프로젝트 <다알아>

김동영 김성진 장창대 조유경

대구 AI 스쿨



## 개요

* 1. 기술 발전에 따라, 소프트웨어의 복잡도는 기하급수적으로 올라가고 있다. 소프트웨어와 더불어 하드웨어 또한 성능이 상승하고 있으며, 최신 소프트웨어는 최신 하드웨어의 기능을 최대한 활용하고 있다. 따라서 개인 사용자의 노후화된 하드웨어로는 최신 소프트웨어를 온전히 활용할 수 없는 경우가 많기 때문에, 개인 사용자들은 주기적으로 하드웨어를 교체하는 경우가 많다.
  2. 그러나 개인 사용자로서는 수많은 종류와 더불어 자주 업그레이드되는 컴퓨터 하드웨어에 대한 전문 지식이 부족하고, 따라서 목적에 맞는 하드웨어 구매를 위해서는 매번 최신 하드웨어에 대한 지식을 공부해야하는 번거로움이 있으며, 지식이 부족할 경우 성능에 따라 가격이 천차만별인 컴퓨터 하드웨어 시장에서 필요 이상의 하드웨어를 구매하게 되는 비효율성이 발생하게 된다
  3. 우리의 목적은 사용자의 컴퓨터 사양을 파악하고, 사용자의 사용 목적을 전달받아 사용 목적의 요구사양과 현재 컴퓨터 사양의 차이를 비교하여, 업그레이드가 필요한 하드웨어를 추천하는 컴퓨터 사양 상담 챗봇을 개발하는 것이다.

## 개발 목표

* 1. 사용자 발화로부터 컴퓨터 사용 목적을 인식하고, 해당 목적에 맞는 컴퓨터 구매 견적을 인터넷 검색한 링크를 제공하는 챗봇 개발

## 프로젝트 깃 허브

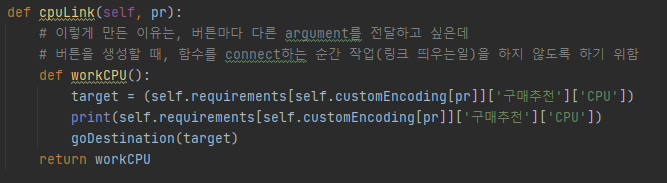
* 1. 주소 : https://github.com/withdongyeong/chatbot
  2. 설치 및 요구사양 등에 대한 명시

## 기능 및 기술 설명

## 

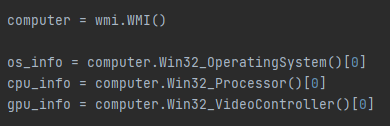
### gui 메신저 형태로 구현

* + 1. PyQt5 라이브러리 사용
    2. 사용자 발화는 텍스트 형태로 입력
    3. 입력된 텍스트를 LSTM 모델로 처리하고, 분류한 후 각 분류에 맞는 각종 api를 호출
    4. 버튼 구성시, 함수를 다음과 같이 구현하였는데 이는 버튼을 생성할 때 기능을 수행하는 것이 아니라, 버튼을 클릭할 때 기능을 수행하기 위함



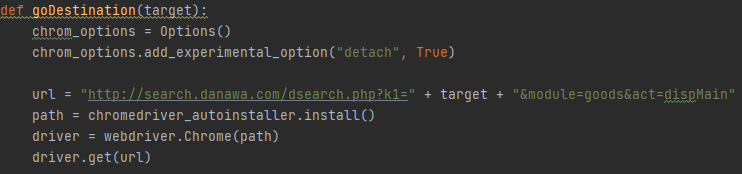
### 사용자 사양 분석

* + 1. wmi 라이브러리를 사용하여 cpu, gpu, ram 사양을 알아낼 수 있다



### 추천 구매 제품 팝업

* + 1. 추천 구매 제품의 검색 결과 링크 url을 브라우저에 팝업한다
    2. selenium 및 chromedriver\_autoinstaller 라이브러리를 사용



### 이해 못한 발화 저장

* + 1. 신뢰도 점수가 낮을 경우 못 알아 들었다고 판단, 이 경우의 사용자 발화를 misunderstanding 발화로 리스트를 만들어 저장한다

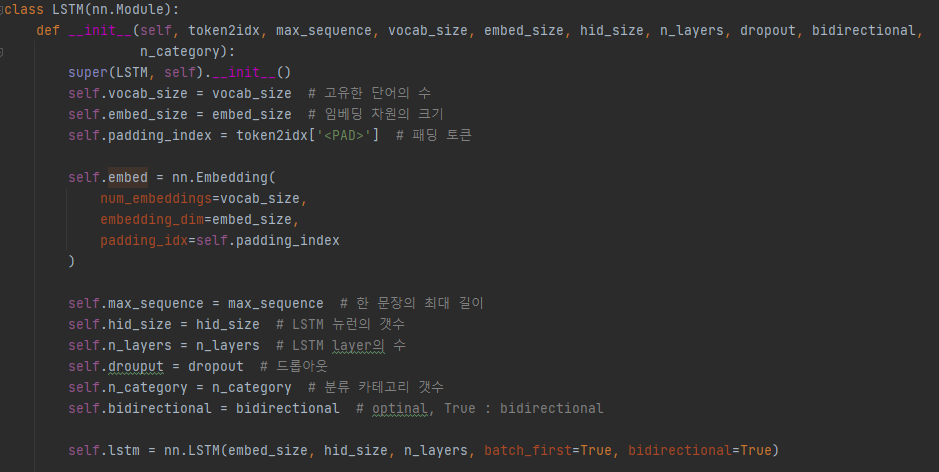
## 데이터 구성

* 1. label 종류는 [롤, 오버워치, 배틀그라운드, 로스트아크, 인사, 감사] 총 6가지
  2. 453개 데이터를 8:2로 학습 및 평가 데이터로 활용

## 데이터 전처리

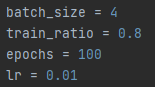
* 1. 자연어 처리를 위해, 각 단어에 정수값을 부여하여 사전을 만들어야 한다
  2. 이 때, 모델에 들어가는 입력값 크기를 맞추기 위해 특정 단어 개수를 기준 개수에 맞춰서 패딩시킨다
  3. 20개를 기준으로 하였으며, 토큰화된 단어의 개수가 20개보다 적을 경우,  
     ‘<PAD>”를 한 단어로 20개 개수를 맞추도록 추가한다
  4. 20개보다 많을 경우, 앞의 20개 단어까지만 인식하고 뒤의 단어는 무시한다
  5. <PAD>의 경우 정수 인덱스를 0을 부여한다

## 모델

* 1. 참고링크 : <https://github.com/BlueWhaleKo/NLP_sentiment_classification>  
     
  2. 해당 프로젝트 내용 중 LSTM 모델을 확장하였음
  3. 변경 사항은
     1. 분류 클래스를 2개에서 6개로 변경
     2. 학습 데이터 포맷(1개의 텍스트파일 -> n개의 텍스트 파일)
     3. 기존 단순 띄어쓰기로 단어를 분리하는데, konlpy의 Komoran을 사용하여 형태소 단위로 단어를 분리하도록 변경
     4. 학습 및 테스트했던 단어사전(int:word dict)에 없는 단어에 대한 예측을 고려하지 않았음
        1. 단어 사전에 없는 단어를 예측하라고 하면, 정수 인덱스를 0을 부여하도록 구현
     5. 매번 예측시 단어사전이 필요한데, 학습 과정을 생략하고 예측하기 위해서 pickle 라이브러리를 사용하여 직렬화, 파일로 저장하였음

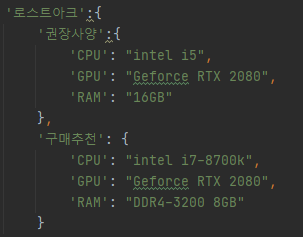


* + 1. 다음 파라미터에 대해 정확도 0.989


## 어려웠던 점

* 1. 원래 기획 의도는 새로 구매/업그레이드 목적을 분리하고, 업그레이드의 경우 사용자의 현재 사양과 권장사양을 비교하여 구매가 필요한 경우에만 해당 하드웨어를 추천하려고 했으나 amd, intel 등의 정량적 비교에 대한 기준을 세워야 해서 보류하였음
  2. 또한 권장사양을 다나와에서 바로 검색시 출시한지 오래된 사양의 경우 완제품이 아니라 해당 하드웨어가 들어간 완성제품이 제시되는 등 검색이 잘 안되서 크롤링 기술을 활용하여 예를 들어 cpu의 경우 cpu 제품 항목을 체크하는 등의 처리를 해야 하나 시간 부족으로 미구현
     1. 따라서 적당히 출시일이 최근인 제품으로 구매 추천사양을 임의로 정의하였음



* 1. konlpy 사용시, 이전 프로젝트에서 잘 사용하였었는데 갑자기 처음 보는 오류가 발생하고, 검색으로 해결이 안 되었는데, 아는 사람이 개발 중에 모듈이 업데이트되어 호환성 문제가 생겼다는 이야기를 한 기억이 나서 확인해보니 konlpy에서 사용하는 tweepy 라이브러리가 21.9.26일에 업데이트되어 4.0이 릴리즈되었는데 이를 konlpy에서 지원하지 않는 문제였다. 따라서 tweepy를 3.7 버전으로 다운그레이드하여 사용하였음