챗봇 프로젝트
다알아

대구 AI 스쿨
김동영 장창대 김성진 조유경

목차

1. 개요
2. 프로젝트 깃 허브
3. 데이터 구성 및 전처리
4. 모델
5. 기능 및 기술 설명
6. 시연
7. 어려웠던 점

1. 개요

기술 발전에 따른 소프트 웨어 복잡도 상승으로, 개인 사용자의 하드웨어를 주기적으로 업그레이드 하지 않으면 최신 소프트웨어를 온전히 활용하기 힘들다.

1. 개요

특히, 최신 게임 소프트웨어 등을 플레이하기 위해 개인 사용자들은 주기적으로 그래픽 카드 등을 업그레이드하는데, 이 때 성능과 가격이 천차만별인 하드웨어 시장에서 명확한 가이드라인이 제공될 필요가 있다.

1. 개요

개발 목표 :

사용자 발화로부터 컴퓨터 사용 목적을 인식하고, 해당 목적에 맞는 컴퓨터 구매 견적을 인터넷에서 검색한 링크를 제공하는 챗봇 개발

2. 프로젝트 깃허브

주소 : https://github.com/withdongyeong/chatbot

설치 및 요구사양 등에 대한 명시

3. 데이터 구성 및 전처리

label 종류 :

[롤, 오버워치, 배틀그라운드, 로스트아크, 인사, 감사] 6가지

데이터 수 : 453개

학습 및 평가 비율 : 8 대 2

3. 데이터 구성 및 전처리

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

3. 데이터 구성 및 전처리

a. 자연어 처리를 위해, 각 단어에 정수값을 부여하여 사전을 만들어야 한다

b. 이 때, 모델에 들어가는 입력값 크기를 맞추기 위해 특정 단어 개수를 기준 개수에 맞춰서 패딩시킨다

c. 20개를 기준으로 하였으며, 토큰화된 단어의 개수가 20개보다 적을 경우,
‘<PAD>”를 한 단어로 20개 개수를 맞추도록 추가한다

3. 데이터 구성 및 전처리

d. 20개보다 많을 경우, 앞의 20개 단어까지만 인식하고 뒤의 단어는 무시한다

e. <PAD>의 경우 정수 인덱스를 0을 부여한다

4. 모델

참고링크 :
https://github.com/BlueWhaleKo/NLP_sentiment_classification

해당 프로젝트 내용 중 LSTM 모델을 확장하였음

4. 모델

```
class LSTM(nn.Module):
    def __init__(self, token2idx, max_sequence, vocab_size, embed_size, hid_size, n_layers, dropout, bidirectional,
                 n_category):
        super(LSTM, self).__init__()
        self.vocab_size = vocab_size  # 고유한 단어의 수
        self.embed_size = embed_size  # 임베딩 차원의 크기
        self.padding_index = token2idx['<PAD>']  # 패딩 토큰

        self.embed = nn.Embedding(
            num_embeddings=vocab_size,
            embedding_dim=embed_size,
            padding_idx=self.padding_index
        )

        self.max_sequence = max_sequence  # 한 문장의 최대 길이
        self.hid_size = hid_size  # LSTM 뉴런의 갯수
        self.n_layers = n_layers  # LSTM layer의 수
        self.drouput = dropout  # 드롭아웃
        self.n_category = n_category  # 분류 카테고리 갯수
        self.bidirectional = bidirectional  # optinal, True : bidirectional

        self.lstm = nn.LSTM(embed_size, hid_size, n_layers, batch_first=True, bidirectional=True)
```

4. 모델

```
# save dict
with open('chatbot.pickle', 'wb') as fw:
    pickle.dump(idx2token, fw)
```

변경사항:

a. 분류 클래스 2개 -> 6개
b. 학습 데이터 포맷
c. 단어 토큰 기준 : 띄어쓰기 -> 형태소(konlpy, Komoran 사용)
d. 학습 및 테스트시 생성한 단어 사전(int:word 형태의 dict)에 없는 단어를 예측하는 경우가 고려되어있지 않아서, 이 경우 정수 0을 부여하도록 구현
e. 매번 예측시 단어사전이 필요한데, 학습 과정을 생략하기 위해 pickle 라이브러리를 사용하여 직렬화하여 파일 저장

4. 모델

다음 파라미터에 대해 정확도 0.989

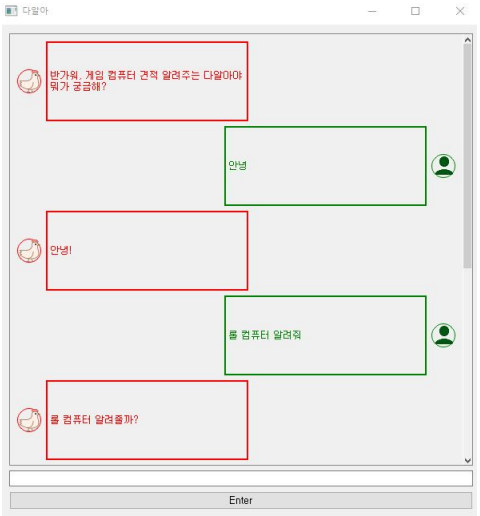
```
batch_size = 4
train_ratio = 0.8
epochs = 100
lr = 0.01
```

```
Test Epoch : 99, Test Loss : 0.014 , Test Accuracy : 0.989
```

5. 기능 및 기술 설명

보고서에 자세히 서술하였으므로 간단히 소개

a. gui 메신저 형태로 구현
   i. PyQt5 라이브러리



5. 기능 및 기술 설명

b. 버튼 생성시 함수, 파라미터를 연결시켰는데 버튼을 생성할 때 호출되지 않고, 버튼을 클릭할 때 호출되도록 하기 위해 다음과 같이 구현

```
for func, buttonName, label in buttonList:
    tempChoiceButton = QPushButton(buttonName)
    tempChoiceButton.clicked.connect(func(label))
```

```
def cpuLink(self, pr):
    # 이렇게 만든 이유는, 버튼마다 다른 argument를 전달하고 싶은데
    # 버튼을 생성할 때, 함수를 connect하는 순간 작업(링크 띄우는일)을 하지 않도록 하기 위함
    def workCPU():
        target = (self.requirements[self.customEncoding[pr]]['구매추천']['CPU'])
        print(self.requirements[self.customEncoding[pr]]['구매추천']['CPU'])
        goDestination(target)
    return workCPU
```

5. 기능 및 기술 설명

c. 사용자 사양 분석
   i. wmi 라이브러리 사용
   ii. 사용자의 cpu, gpu, ram 사양 분석

```
computer = wmi.WMI()

os_info = computer.Win32_OperatingSystem()[0]
cpu_info = computer.Win32_Processor()[0]
gpu_info = computer.Win32_VideoController()[0]
```

5. 기능 및 기술 설명

d. 추천 구매 제품 팝업
   i. selenium 및 chromedriver_autoinstaller 라이브러리 사용
   ii. 추천 구매 제품의 검색 결과 링크를 브라우저에 팝업

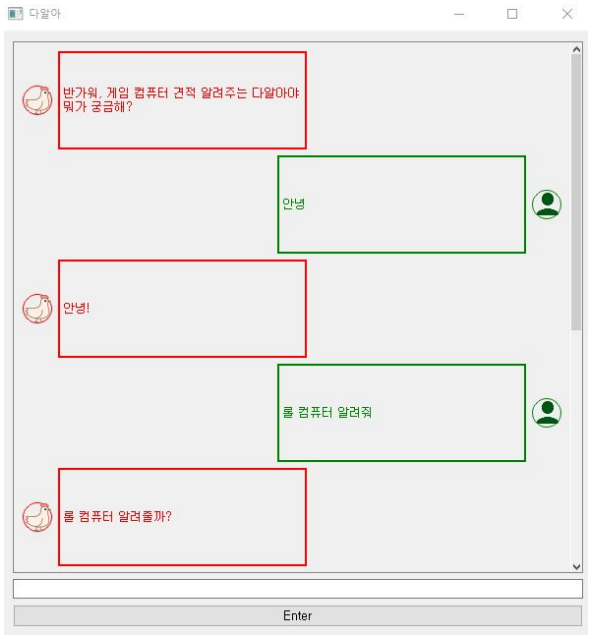
```
def goDestination(target):
    chrom_options = Options()
    chrom_options.add_experimental_option("detach", True)

    url = "http://search.danawa.com/dsearch.php?k1=" + target + "&module=goods&act=dispMain"
    path = chromedriver_autoinstaller.install()
    driver = webdriver.Chrome(path)
    driver.get(url)
```

5. 기능 및 기술 설명

e. 이해 못한 발화 저장
   i. 신뢰도 점수가 낮을 경우, 못 알아 들었다고 판단하고 이 발화들을 텍스트 파일로 저장

6. 시연


다알아
반가워, 게임 컴퓨터 견적 알려주는 다알아야
뭐가 궁금해?
안녕
안녕!
롤 컴퓨터 알려줘
롤 컴퓨터 알려줄까?
Enter

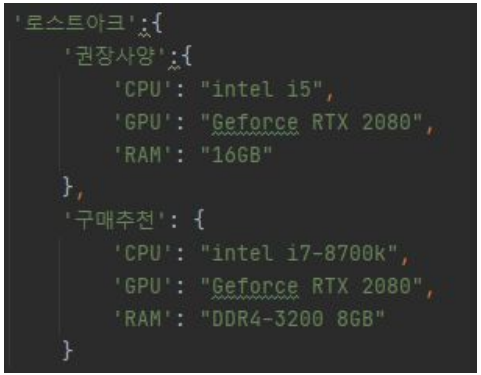
7. 어려웠던 점

원래 기획 의도:

새로 구매/업그레이드 목적을 분리,
업그레이드의 경우 사용자의 현재 사양과 권장사양을
비교하여 구매가 필요한 경우에만 해당 하드웨어를
추천하려고 했으나
amd, intel 등의 정량적 비교에 대한 기준을 세워야하는 문제가
있었고, 충분히 고려하지 못 해서 보류하였음

7. 어려웠던 점

추천 제품을 다나와에서 바로 검색할 경우,
특히 출시한지 오래된 제품의 경우 예를 들어
cpu 완제품이 검색되는 것이 아니라 해당 cpu가 들어있는
노트북이 검색되는 문제



-> 크롤링 기술을 적용하여 cpu 완제품에 체크를 하는 등의
처리를 해야했으나 시간 부족으로 미구현

-> 적당히 출시일이 최근인 제품으로 구매 추천 사양을
정의하였음

7. 어려웠던 점

konlpy를 사용하여 모델 구현을 9.27일에 작업하는 도중, 이전 프로젝트에서는 잘 사용하였음에도 처음 보는 오류가 발생하고 검색으로 해결이 안 되었음.

아는 사람이 개발 중에 모듈이 업데이트되어 호환성 문제가 생겼다는 이야기를 한적이 있어서, 체크해보니

konlpy의 하위 모듈 tweepy가 9.26에 최신 버전이 릴리즈되었음. 이를 지원하지 않으므로 다운 그레이드해서 사용