Information Search Project

B813005 고민재

INDEX

주제 선정 과정

저는 아직 뭐가 뭔지 모르겠어요ㅠㅠㅠ

최근 캐글 데이터캠프에 참석

캐글에서 데이터와 주제를 찾아보자

뭘 해야 하는 걸까 나는...

혼자 할 수 있는 걸까...

연세대 컴퓨터과학과 친구의

친구의 친구의... NLP 활용 프로젝트 소개

주제 결정

에브리타임 홍.맛.게 Sentiment Analysis

사람들이 쓴 글을 분석해서 긍정후기와 부정후기로 분류하는 딥러닝을 해보자

대학생만을 위한 커뮤니티

에브리타임

에브리타임 <야 넌 홍대생인데 홍대맛집도 모르냐-를 방지하기 위한 홍대맛집게시판> 크롤링

Keras RNN으로 네이버 영화 리뷰 감성 분석

Hugging Face의 PyTorch BERT로 모델 만들어 전이 학습

머신러닝 아키텍처

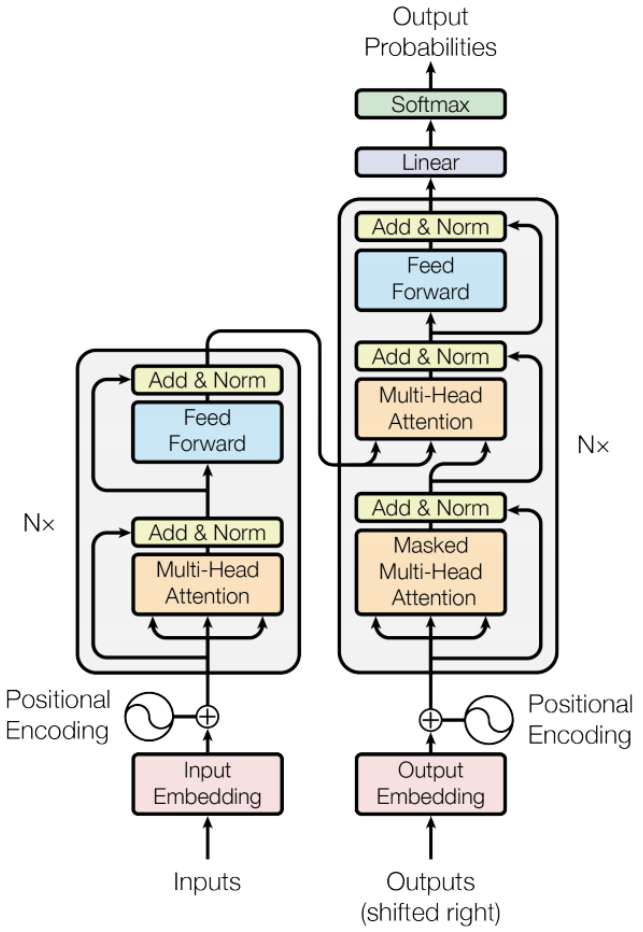
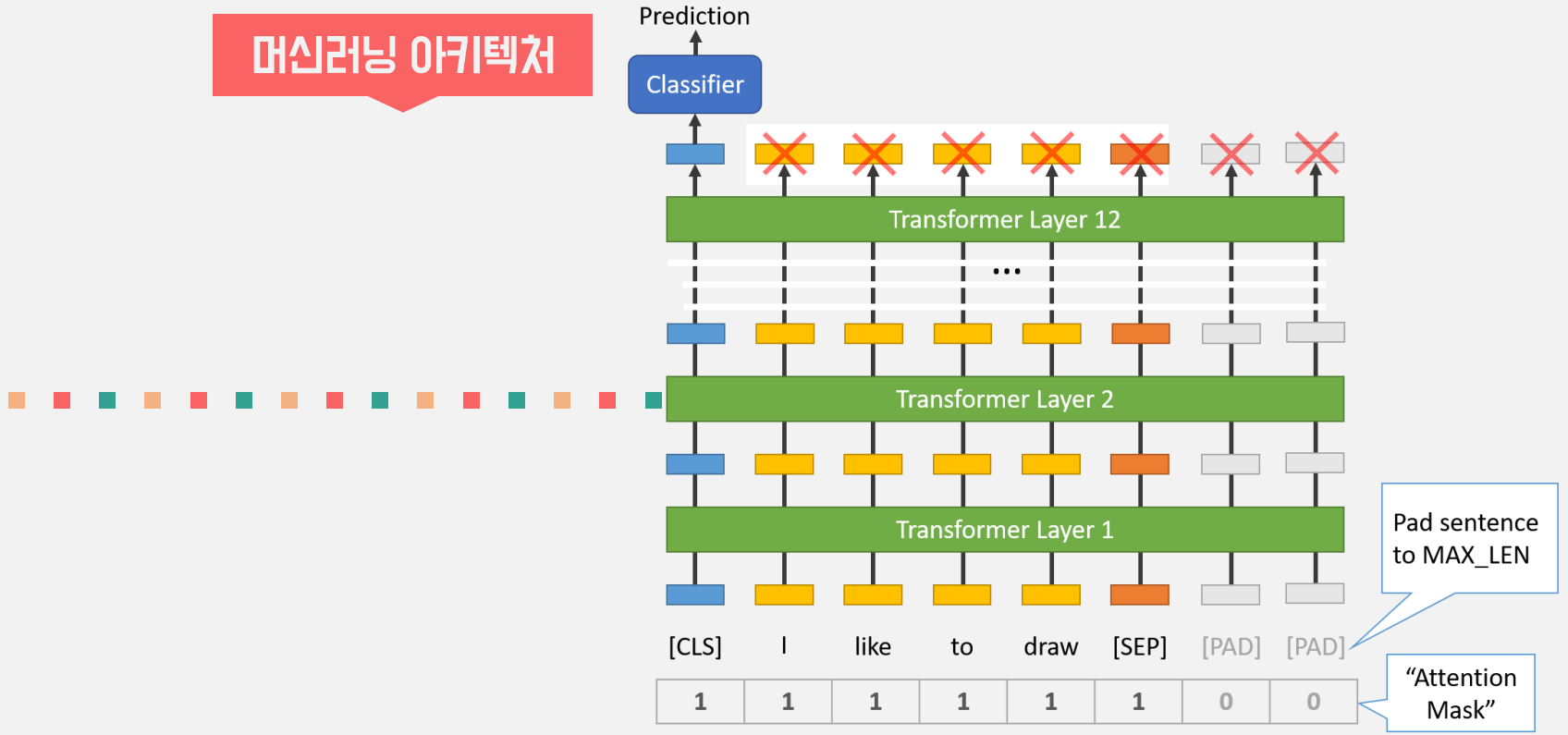


Figure 1: The Transformer - model architecture.



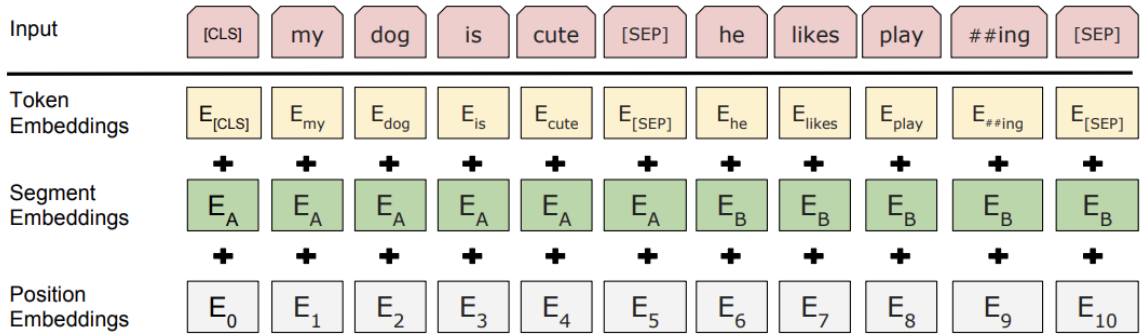
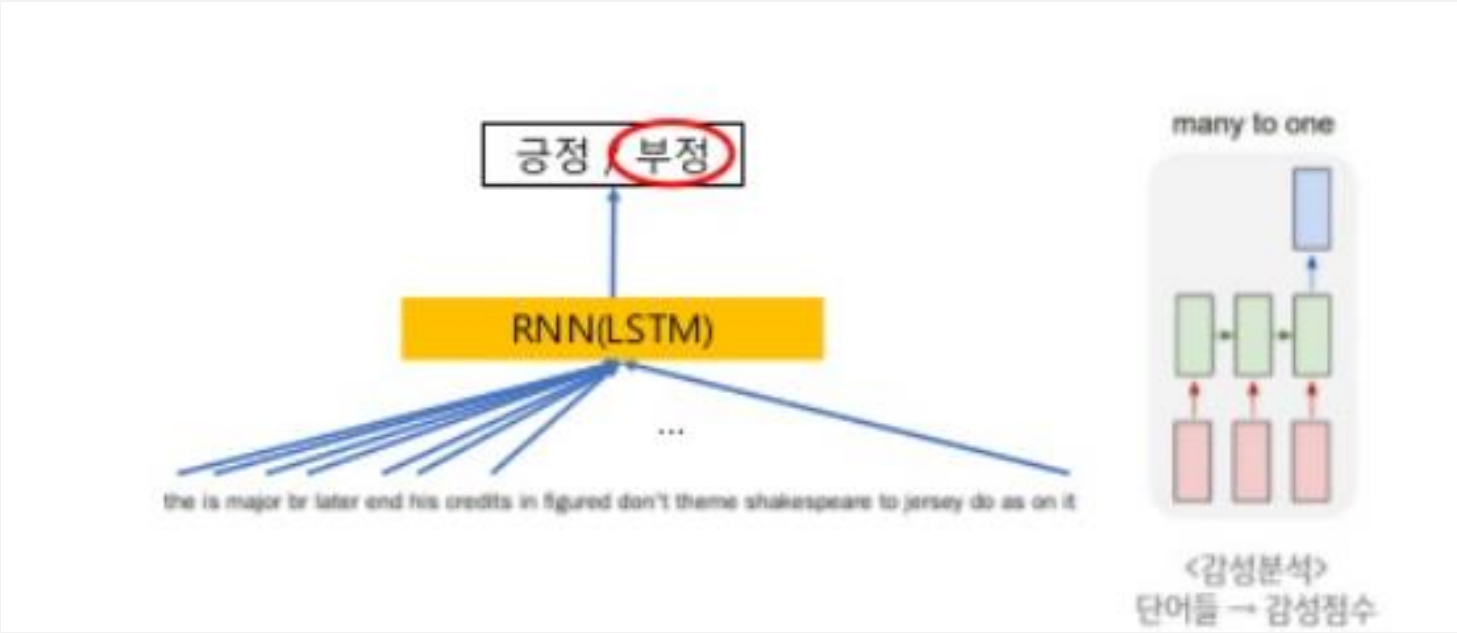


Figure 2: BERT input representation. The input embeddings is the sum of the token embeddings, the segmentation embeddings and the position embeddings.

머신러닝 아키텍처


긍정 / 부정
RNN(LSTM)
...
the is major br later end his credits in figured don't theme shakespeare to jersey do as on it
many to one
<감성분석>
단어들 → 감성점수

활용 데이터와 계획

```
#엑셀 저장화
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
from urllib.request import urlopen
from bs4 import BeautifulSoup
import time
from konlpy.tag import Kkma
import openpyxl
#라이브러리 불러오기

kma = Kkma()

excel_file = openpyxl.Workbook()
excel_sheet = excel_file.active
#엑셀 파일로 저장하기 위한 작업 s

chromedriver ='C:\dev_python\Web
driver = webdriver.Chrome(chrome
#selenium 라이브러리 기본 작업

driver.get('https://everytime.kr
#크롤링 할 홈페이지 가져오기
```

Naver sentiment movie corpus v1.0

This is a movie review dataset in the Korean language. Reviews were scraped from Naver Movies.

The dataset construction is based on the method noted in Large movie review dataset from Maas et al., 2011.

Data description

- Each file is consisted of three columns: `id`, `document`, `label`
  - `id` : The review id, provieded by Naver
  - `document` : The actual review
  - `label` : The sentiment class of the review. (0: negative, 1: positive)
  - Columns are delimited with tabs (i.e., `.tsv` format; but the file extension is `.txt` for easy access for novices)
- 200K reviews in total
  - `ratings.txt` : All 200K reviews
  - `ratings_test.txt` : 50K reviews held out for testing
  - `ratings_train.txt` : 150K reviews for training

크롤러제작 및 데이터 분류 수집 ~11/17
코드 구현 ~11/22
모델링 완성 및 분석 완료 ~11/29

성능 평가 방법

테스트셋 평가

```
#시작 시간 설정
t0 = time.time()

# 평가모드로 변경
model.eval()

# 변수 초기화
eval_loss, eval_accuracy = 0, 0
nb_eval_steps, nb_eval_examples = 0, 0

# 데이터로더에서 배치만큼 반복하여 가져옴
for step, batch in enumerate(test_dataloader):
    # 경과 정보 표시
    if step % 100 == 0 and not step == 0:
        elapsed = format_time(time.time() - t0)
        print('  Batch {:>5,}  of  {:>5,}.    Elapsed: {:}.'.format(step, len(test_dataloader), elapsed))

    # 배치를 GPU에 넣음
    batch = tuple(t.to(device) for t in batch)

    # 배치에서 데이터 추출
    b_input_ids, b_input_mask, b_labels = batch

    # 그래디언트 계산 안함
    with torch.no_grad():
        # Forward 수행
        outputs = model(b_input_ids,
                        token_type_ids=None,
                        attention_mask=b_input_mask)

    # 로스 구함
    logits = outputs[0]

    # CPU로 데이터 이동
    logits = logits.detach().cpu().numpy()
    label_ids = b_labels.to('cpu').numpy()

    # 출력 로짓과 라벨을 비교하여 정확도 계산
    tmp_eval_accuracy = flat_accuracy(logits, label_ids)
    eval_accuracy += tmp_eval_accuracy
    nb_eval_steps += 1

print("")
print("Accuracy: {0:.2f}".format(eval_accuracy/nb_eval_steps))
print("Test took: {:}".format(format_time(time.time() - t0)))
```

```
# 정확도 계산 함수
def flat_accuracy(preds, labels):

    pred_flat = np.argmax(preds, axis=1).flatten()
    labels_flat = labels.flatten()

    return np.sum(pred_flat == labels_flat) / len(labels_flat)
```

```
  Batch   100  of  1,563.    Elapsed: 0:00:12.
  Batch   200  of  1,563.    Elapsed: 0:00:25.
  Batch   300  of  1,563.    Elapsed: 0:00:37.
  Batch   400  of  1,563.    Elapsed: 0:00:49.
  Batch   500  of  1,563.    Elapsed: 0:01:01.
  Batch   600  of  1,563.    Elapsed: 0:01:14.
  Batch   700  of  1,563.    Elapsed: 0:01:26.
  Batch   800  of  1,563.    Elapsed: 0:01:38.
  Batch   900  of  1,563.    Elapsed: 0:01:50.
  Batch 1,000  of  1,563.    Elapsed: 0:02:03.
  Batch 1,100  of  1,563.    Elapsed: 0:02:15.
  Batch 1,200  of  1,563.    Elapsed: 0:02:27.
  Batch 1,300  of  1,563.    Elapsed: 0:02:39.
  Batch 1,400  of  1,563.    Elapsed: 0:02:52.
  Batch 1,500  of  1,563.    Elapsed: 0:03:04.

Accuracy: 0.87
Test took: 0:03:11
```

활용자료

https://github.com/deepseasw/bert-naver-movie-review
https://github.com/snoop2head/yonsei-exchange-program
https://wikidocs.net/44249
https://blog.yourssu.com/post/everytime-bot-1/
https://itsallgoodman.tistory.com/9

Thank You