

```

import torch
import gluonnlp as nlp
import numpy as np
import random
#kobert
from kobert.utils import get_tokenizer
from kobert.pytorch_kobert import get_pytorch_kobert_model
from bertDataset import BERTDataset

class Emotion:
    def __init__(self):
        self.max_len = 100
        self.batch_size = 16
        bertmodel, vocab = get_pytorch_kobert_model()
        #토큰화
        tokenizer = get_tokenizer()
        self.tok = nlp.data.BERTSPTokenizer(tokenizer, vocab, lower=False)

    def emotion(self):
        max_len = 100
        batch_size = 16
        bertmodel, vocab = get_pytorch_kobert_model()
        #토큰화
        tokenizer = get_tokenizer()
        tok = nlp.data.BERTSPTokenizer(tokenizer, vocab, lower=False)
        device = torch.device("cuda:0")

        with torch.no_grad():
            end = 1
            while end == 1 :
                sentence = input("지금 하고싶은 말을 해줘"+'\n')
                model = torch.load('C:/MyProject/chatbot/save/chatbot_v50.pth')
                data = [sentence, '0']
                dataset_another = [data]

                another_test = BERTDataset(dataset_another, 0, 1, tok, max_len, True, False)
                test_dataloader = torch.utils.data.DataLoader(another_test, batch_size=batch_size, num_workers=0)

                model.eval()

                for batch_id, (token_ids, valid_length, segment_ids, label) in enumerate(test_dataloader):
                    token_ids = token_ids.long().to(device)
                    segment_ids = segment_ids.long().to(device)

                    valid_length= valid_length
                    label = label.long().to(device)

                    out = model(token_ids, valid_length, segment_ids)
                    # 힘들고 지칠 때 읽으면 좋은 책
                    sad_books = ['나는 나로 살기로 했다', '단상집', '그냥 올려놔어도 좋아요', '아무것도 안 해도 아무렇지 않구나', '죽고 싶지만 떡볶이는 먹고 싶어', '내가 이
                    happy_books = ['불편한 편의점', '어른을 위한 인생수업', '봄이다, 살아보자', '우리는 숲으로 여행간다', '봄의 초대', '입지 센스', '파친코', '아몬드', '
                    angry_books = ['3초간', '나는 오늘부터 화를 끊기로 했다.', '오늘도 욕하셨나요?', '디퓨징', '오늘도 화를 내고 말았습니다', '용서', '화, 참을 수 없다면

                test_eval=[]
                for i in out:
                    logits=i
                    logits = logits.detach().cpu().numpy()

                    if np.argmax(logits) == 0: # 화남
                        test_eval.append("마음을 가라앉히고 싶을 때는 ")
                    elif np.argmax(logits) == 1: # 슬픔
                        test_eval.append("마음의 위로가 필요할 때는 ")
                    elif np.argmax(logits) == 2: # 행복
                        test_eval.append(" ")

                if test_eval[0] == "마음을 가라앉히고 싶을 때는 ":
                    print(">> " + test_eval[0] + random.choice(angry_books) + " 이 책을 읽어보세요")
                elif test_eval[0] == "마음의 위로가 필요할 때는 ":
                    print(">> " + test_eval[0] + random.choice(sad_books) + " 이 책을 읽어보세요")
                elif test_eval[0] == " ":
                    print(">> " + test_eval[0] + random.choice(happy_books) + " 이 책을 읽어보세요")

            break

```