

## main.py

```
import tensorflow as tf

import model as ml

import data

import numpy as np

import os

import sys

from configs import DEFINES

DATA_OUT_PATH = './data_out/'

data_out_path = os.path.join(os.getcwd(), DATA_OUT_PATH)
os.makedirs(data_out_path, exist_ok=True)

# 데이터를 통한 사전 구성 한다.

# 질문과 응답 문장의 단어를 형태소로 변환하고, 단어 목록 dict 를 만든다.
word2idx, idx2word, vocabulary_length = data.load_vocabulary()

# 훈련 데이터와 테스트 데이터를 가져온다.

# 질문과 응답 문장을 학습 데이터와 시험 데이터로 분리한다.

# ipdb> train_input[0]

# '짝사랑 했던 여자가 떠나갔네.'

# ipdb> train_label[0]

# '여기까지 인연이었나봅니다.'

train_input, train_label, eval_input, eval_label = data.load_data()

# 훈련셋 인코딩 / 디코딩 입력 / 디코딩 출력 만드는 부분이다.

#ipdb> train_input_enc[0]

#array([107,  86,  40,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
#        0,   0,   0,   0,   0,   0,   0,   0,   0,   0])
```

```

#ipdb> train_input_dec[0]
#array([ 1, 53, 14, 23, 48,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
#        0,  0,  0,  0,  0,  0,  0,  0])
#ipdb> train_target_dec[0]
#array([53, 14, 23, 48,  2,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
#        0,  0,  0,  0,  0,  0,  0,  0])
train_input_enc, train_input_enc_length = data.enc_processing(train_input, word2idx)
train_input_dec, train_input_dec_length = data.dec_input_processing(train_label,
word2idx)
train_target_dec = data.dec_target_processing(train_label, word2idx)

# 평가셋 인코딩 / 디코딩 입력 / 디코딩 출력 만드는 부분이다.
eval_input_enc, eval_input_enc_length = data.enc_processing(eval_input, word2idx)
eval_input_dec, eval_input_dec_length = data.dec_input_processing(eval_label,
word2idx)
eval_target_dec = data.dec_target_processing(eval_label, word2idx)

# 현재 경로 './'에 현재 경로 하부에 체크 포인트를 저장한 디렉토리를 설정한다.
check_point_path = os.path.join(os.getcwd(), DEFINES.check_point_path)
os.makedirs(check_point_path, exist_ok=True)

# 에스티메이터를 구성한다.
classifier = tf.estimator.Estimator(
    model_fn=m1.model, # 모델 등록한다.
    model_dir=DEFINES.check_point_path,
    params={
        'hidden_size': DEFINES.hidden_size,
        'layer_size': DEFINES.layer_size,
        'learning_rate': DEFINES.learning_rate,
        'vocabulary_length': vocabulary_length,
        'embedding_size': DEFINES.embedding_size,
        'embedding': DEFINES.embedding,
        'multilayer': DEFINES.multilayer,
    })

```

# 학습 실행

```
classifier.train(input_fn=lambda: data.train_input_fn(  
    train_input_enc, train_input_dec, train_target_dec,    DEFINES.batch_size),  
steps=DEFINES.train_steps)
```

# 평가 실행

```
eval_result = classifier.evaluate(input_fn=lambda: data.eval_input_fn(  
    eval_input_enc, eval_input_dec, eval_target_dec, DEFINES.batch_size))  
print('\nEVAL set accuracy: {accuracy:0.3f}\n'.format(**eval_result))
```

# 테스트셋 인코딩 / 디코딩 입력 / 디코딩 출력 만드는 부분이다.

```
predic_input_enc, predic_input_enc_length = data.enc_processing(["가끔 궁금해"],  
word2idx)  
predic_input_dec, predic_input_dec_length = data.dec_input_processing([""],  
word2idx)  
predic_target_dec = data.dec_target_processing([""], word2idx)
```

# 예측 실행

```
predictions = classifier.predict(  
    input_fn=lambda: data.eval_input_fn(predic_input_enc, predic_input_dec,  
predic_target_dec, DEFINES.batch_size))
```

# 예측한 값을 텍스트로 변경하는 부분이다.

```
data.pred2string(predictions, idx2word)
```