과제 4: LSTM모델 기반 영어 품사인식 시스템 개발

• 과제 내용: 영어 품사인식 시스템을 개발하고 실험한다. 제공되는 guide 프로그램을 확장하여 개발하도록 한다. 제공된 프로그램의 진행은 다음과 같다.

1) 작업을 위해 영어문장의 각 단어에 품사명을 붙인 훈련데이타 파일 all_word_pos_sentences_all.txt 이 주어진다. 그리고 이 파일을 3 개의 부분으로 나누어 놓은 다음 3 개의 파일도 주어진다:

all_word_pos_sentences_train.txt,
all_word_pos_sentences_validation.txt,
all_word_pos_sentences_test.txt

2) 단어사전 및 품사사전 만들기 (제공한 guide 프로그램 참고)
- all_word_pos_sentences_all.txt 에 등장하는 모든 단어를 수집하여 단어사전을 만든다. 이 사전에는 두 개의 특수단어 [PAD], [UNK] 도 추가한다. 사전명: Vocab
  이 사전은 python 의 dict 타입을 이용한다. key 는 단어, value 는 단어번호이다.
  단어의 총수는 51,459 개이다.
  역단어사전도 만든다: i_Vocab
- 품사사전 만들기: 사전명: dic_POS, key:품사명, value: 품사번호
  0 번은 [PAD]의 품사, 1 번은 [UNK] 의 품사이다. 나머지 48 개는 penn-tree-bank 품사표의 것들이다.
  역품사사전도 만든다: i_dic_POS

3) 단어/품사명으로 된 파일로부터 단어번호/품사번호으로 된 파일을 생성한다(제공한 guide 프로그램 참고)

all_word_pos_sentences_train.txt ==> all_index_sentences_train.txt
all_word_pos_sentences_validation.txt ==> all_index_sentences_train.txt
all_word_pos_sentences_test.txt ==> all_index_sentences_train.txt

```
build_index_sentences("./drive/MyDrive/all_word_pos_sentences_train.txt", "./drive/MyDrive/all_index_sentences_train.txt")
build_index_sentences("./drive/MyDrive/all_word_pos_sentences_validation.txt",
"./drive/MyDrive/all_index_sentences_validation.txt")
build_index_sentences("./drive/MyDrive/all_word_pos_sentences_test.txt", "./drive/MyDrive/all_index_sentences_test.txt")
```

4) 단어번호/품사번호로 된 파일로 부터 training examples 를 준비한다.
이를 위해 제공한 함수 load_X_and_Y 을 이용한다.

```
x_train, y_train, leng_train = load_X_and_Y("./drive/MyDrive/all_index_sentences_train.txt")
x_train = np.array(x_train, dtype='i')
y_train = np.array(y_train, dtype='i')

x_validation, y_validation, leng_valiation = load_X_and_Y("./drive/MyDrive/all_index_sentences_validation.txt")
x_validation = np.array(x_validation, dtype='i')
y_validation = np.array(y_validation, dtype='i')

x_test, y_test, leng_test = load_X_and_Y("./drive/MyDrive/all_index_sentences_test.txt")
x_test = np.array(x_test, dtype='i')
y_test = np.array(y_test, dtype='i')
```

5) 모델 구축 및 훈련

```
# Model 설계

model = tf.keras.models.Sequential()

## 층0: word-embedding 층
model.add(tf.keras.layers.Embedding(Vocab_size, d_embed, embeddings_initializer='random_normal', W
 input_length=Max_seq_length, mask_zero=True, trainable=True))         # output shape: (bsz, MSL, d_emb)

## 층1: LSTM 층
model.add(tf.keras.layers.Bidirectional(tf.keras.layers.LSTM(256,      return_sequences=True),      input_shape=(Max_seq_length,
d_embed)))
 # output shape: (batch_sz, MSL, 512)
```

```
## 층2: LSTM 층
model.add(tf.keras.layers.Bidirectional(tf.keras.layers.LSTM(128, return_sequences=True), input_shape=(Max_seq_length, 512)))
 # output shape: (batch_sz, MSL, 256)

## 층3: LSTM 층
model.add(tf.keras.layers.Bidirectional(tf.keras.layers.LSTM(64, return_sequences=True), input_shape=(Max_seq_length, 256)))
 # output shape: (batch_sz, MSL, 128)

## 층4: NN 층
model.add(tf.keras.layers.Dense(units=Num_POS, activation='softmax', use_bias=True))      # 최종출력층. 각 시간의 각 단어마다 num_POS=50개의 확률이 생성됨.
 # output shape: (batch_sz, MSL, num_POS)

optimizer = tf.keras.optimizers.Adam(learning_rate=LEARNING_RATE)
model.compile(optimizer=optimizer, loss="sparse_categorical_crossentropy", metrics=['acc'])
model.fit(x=x_train,   y=y_train,   batch_size=BATCH_SIZE,   epochs=EPOCHS,   validation_data=(x_validation,   y_validation), shuffle=True, verbose=1)
```

6) test 실험

(6-1) test 훈련예제들을 이용하여 모델에 의한 예측(prediction) 작업을 수행한다.
이를 위해 다음과 같은 코드를 수행한다.

```
pred = model.predict(x=x_test, verbose=1)
pred_label = tf.math.argmax(pred, axis=2)
pred_label = pred_label.numpy()     # this is of 2-dimension (num_test_sents, msl)
num_test_sentences = y_test.shape[0]  # y_test has a shape of (num_test_sents, msl)
```

위 prediction 작업의 결과를 사용하여 x_test 의 각 예제마다 다음 작업을 수행한다:
    단어마다 모델의 예측이 올바른 지를 판단한다(y_test 의 정답 정보 이용).
    여기서 주의할 점은 leng_test 에 있는 문장의 길이 즉 단어 수 만큼만 판단해야 한다
     (즉 padding 을 채운 시간에 대해서는 예측이 정답여부 판단을 하지 않아야 한다).
    이를 근거로 하여 test 데이타에 대한 모델의 품사인식 작업의 accuracy 를 측정한다.
이를 위한 코드를 작성하여 넣고 실행하여 accuracy 를 화면에 출력하도록 한다.

(6-2) 품사인식 결과 출력

첫 test 예제 20 문장에 대하여 예측작업의 결과를 다음처럼 출력한다.
여기서 단어에 대하여 품사인식이 올바르면 인식한 올바른 품사만 붙여 주고, 품사인식이 틀렸으면 잘못 인식한 품사및 정답 품사를 같이 붙여준다(정답품사는 각괄호에 넣어서 붙임). 예를 들어 설명하자.

    (1) you/PR watches/NN<VBZ> the/DT shows/VBZ at/IN the/DT balcony/NN
    (2) Mr./NNP Goldberg/NNP is/VBZ the/DT sole/JJ general/NN<JJ> partner/NN in/IN
          Rose/NNP Partners/NNP ./.
    (3) ....

위에서 단어 you 옆에 /PR 은 you 의 품사를 PR 로 올바로 인식했음을 나타낸다.
반면에 watches 의 경우 /NN 은 모델이 NN 으로 인식하였는데 이는 잘못 인식한 결과로서, 실제로는 < > 안의 VBZ 가 정답품사임을 나타낸다.

(힌트: 역사전을 이용하여 단어번호를 단어로, 품사번호를 품사명으로 바꾸어 주어야 한다.)

- 제출물:
  - 프로그램 (ipynb 파일)
  - 보고서
    보고서에는 다음 사항을 넣는다:
    (6-1) 단계에서의 test accuracy 화면 출력 부분을 캡쳐하여 넣는다.
    (6-2) 단계에서의 10 문장에 대한 화면 출력 부분을 캡쳐하여 넣는다.