# [HW5] Bi-directional LSTM for Named Entity Recognition

120180048

김태훈

1. Source code

   i. json2CoNLL.py

   json2CoNLL.py의 converter 함수는 data 폴더 내에 있는 json 파일을 읽어 CoNLL 형태로 바꿔주는 작업을 수행한다.

```
rawdata = json.loads(open(filename,'rt',encoding='utf-8-sig').read())
newfile = '../data/NER_'+filename.split('_')[-1].split('.')[0]+'.txt'
wfp = open(newfile,'w')
sentence_id = len(rawdata['sentence'])
```

   우선 이와 같은 방식으로 python 내의 json 라이브러리를 이용하여 json 파일을 읽는다.

```
        for j in range(morp_id):
            cur_morp = sentence['morp'][j]
            if cur_word_idx+1 < word_len:
                if (cur_morp['lemma'] not in sentence['word'][cur_word_idx]['text']) and
(cur_morp['lemma'] in sentence['word'][cur_word_idx+1]['text']):
                    cur_word_idx += 1
            if cur_NE_idx+1 < NE_len:
                if (cur_morp['lemma'] not in sentence['NE'][cur_NE_idx]['text']) and (cur_morp['lemma']
in sentence['NE'][cur_NE_idx+1]['text']):
                    cur_NE_idx += 1

            cur_word = sentence['word'][cur_word_idx]
            cur_NE = sentence['NE'][cur_NE_idx]
            if cur_word['text'] == cur_NE['text']:
                if cur_morp['lemma'][0] == cur_word['text'][0]:
                    bio = 'B'
                else:
                    bio = 'I'
                tag = bio+'-'+cur_NE['type']
            else:
                tag = 'O'
            print(cur_morp['lemma']+'/'+cur_morp['type']+' '+tag, file=wfp)
```

   이후 json 파일 내 각각의 sentence에 대하여 해당 sentence 내의 형태소를 모두 태깅하는 작업을 수행하게 된다. 이 과정에서 cur_word_idx를 이

용하여 현재 태깅 중인 형태소가 위치한 word를 인덱싱하고, 해당 word에 대해서 태깅된 Named Entitiy가 있는지 확인하기 위하여 추가적으로 cur_NE_idx를 사용하였다.

이를 통해 현재 작업 중인 형태소가 포함된 word에 해당하는 NE가 없을 경우에는 해당 형태소를 O로 태깅하고, 있을 경우에는 해당 형태소가 word의 Begin인지 Inside인지를 추가적으로 확인하여 해당 형태소에 알맞은 NE를 태깅하는 작업을 수행한다.

최종 결과 파일은 data 디렉토리 내에 저장된다.

ii. model.py

* 강의 자료에 자세히 설명되어 있는 내용에 대해서는 설명을 생략

- Bidirectional dynamic RNN을 이용한 LSTM 모델 구축

```
cell_fw = tf.contrib.rnn.LSTMCell(self.config.hidden_size_lstm)
cell_bw = tf.contrib.rnn.LSTMCell(self.config.hidden_size_lstm)
(output_fw, output_bw), _ = tf.nn.bidirectional_dynamic_rnn(
            cell_fw, cell_bw, word_embeddings, self.sequence_lengths, dtype=tf.float32)
```

- 출력 레이어 정의

```
output = tf.reshape(output, [-1, 2*self.config.hidden_size_lstm])
W = tf.get_variable("W", dtype=tf.float32,
                    shape=[2*self.config.hidden_size_lstm, self.config.ntags])
b = tf.get_variable("b", dtype=tf.float32,
                    shape=[self.config.ntags],initializer=tf.zeros_initializer())
pred = tf.matmul(output, W)+b
```

- 음절 자질 추가를 위한 placeholder 추가 생성

```
self.word_lengths= tf.placeholder(tf.int32, shape=[None,None], name="word_lengths")
self.char_ids = tf.placeholder(tf.int32, shape=[None,None,None], name="char_ids")
```

- character embedding 생성

```
with tf.variable_scope("chars"):
    print("Randomly initializing char vectors")
    _char_embeddings = tf.get_variable(
            name="_char_embeddings",
            dtype=tf.float32,
            shape=[self.config.nchars, self.config.dim_char])
    char_embeddings = tf.nn.embedding_lookup(_char_embeddings,
            self.char_ids, name="char_embeddings")
```

- character embedding을 위한 bi-LSTM 네트워크 생성

```
dim_for_rnn = tf.shape(char_embeddings)
char_embeddings = tf.reshape(char_embeddings,
        shape=[dim_for_rnn[0]*dim_for_rnn[1], dim_for_rnn[-2], self.config.dim_char])
word_lengths = tf.reshape(self.word_lengths, shape=[dim_for_rnn[0]*dim_for_rnn[1]])

cell_fw = tf.contrib.rnn.LSTMCell(self.config.hidden_size_char)
cell_bw = tf.contrib.rnn.LSTMCell(self.config.hidden_size_char)
_output = tf.nn.bidirectional_dynamic_rnn(
        cell_fw, cell_bw, char_embeddings, word_lengths, dtype=tf.float32)

_, ((_, output_fw), (_, output_bw)) = _output
output = tf.concat([output_fw, output_bw], axis=-1)
```

이 과정에서 character embedding이 RNN 모델에 들어갈 수 있도록 [ batch_size, sequence_length, word_length ] 형태인 char_embeddings를 [ batch_size * sequence_length, word_length ]의 형태로 변형하였으며, word_lengths placeholder 역시 [ batch_size * sequence_length]의 형태로 변형시켜 주었다.

- bi-LSTM을 통과한 음절 표현을 기존의 단어 표현에 결합

```
_, ((_, output_fw), (_, output_bw)) = _output
output = tf.concat([output_fw, output_bw], axis=-1)

output = tf.reshape(output,
        shape=[dim_for_rnn[0], dim_for_rnn[1], 2*self.config.hidden_size_char])
word_embeddings = tf.concat([word_embeddings, output], axis=-1)
```

2. Randomly initialized embedding을 이용한 실험 결과

i. 숙제 2

[ train.py ]

```
Epoch 1 out of 10
acc : 96.79 - f1 : 1.31
new best score
Epoch 2 out of 10
acc : 97.09 - f1 : 28.80
new best score
Epoch 3 out of 10
acc : 97.19 - f1 : 40.89
new best score
Epoch 4 out of 10
acc : 97.27 - f1 : 42.61
new best score
Epoch 5 out of 10
acc : 96.85 - f1 : 46.09
new best score
Epoch 6 out of 10
acc : 96.31 - f1 : 43.96
Epoch 7 out of 10
acc : 96.77 - f1 : 45.94
Epoch 8 out of 10
acc : 97.23 - f1 : 43.08
- early stopping 3 epochs without improvement
```

[ evaluate.py ]

```
Reloading the latest trained model...
Testing model over test set
acc : 96.86 - f1 : 39.20
23/SN B-DT
일/NNB I-DT
기성용/NNP O
의/JKG O
활약/NNG O
으로/JKB O
스완지시티/NNP O
는/JX O
리버풀/NNP O
선/NNG O
에서/JKB O
승리/NNG O
를/JKO O
얻/VV O
었/EP O
다/EC O
./SF O
```

ii. 숙제 3

[ train.py ]

```
Epoch 1 out of 10
acc : 96.91 - f1 : 12.77
new best score
Epoch 2 out of 10
acc : 97.10 - f1 : 27.79
new best score
Epoch 3 out of 10
acc : 97.21 - f1 : 34.30
new best score
Epoch 4 out of 10
acc : 97.27 - f1 : 52.48
new best score
Epoch 5 out of 10
acc : 97.02 - f1 : 50.96
Epoch 6 out of 10
acc : 97.27 - f1 : 49.29
Epoch 7 out of 10
acc : 97.23 - f1 : 43.26
- early stopping 3 epochs without improvement
```

[ evaluate.py ]

```
Reloading the latest trained model...
Testing model over test set
acc : 97.14 - f1 : 45.86
23/SN B-DT
일/NNB I-DT
기성용/NNP O
의/JKG O
활약/NNG O
으로/JKB O
스완지시티/NNP O
는/JX O
리버풀/NNP O
선/NNG O
에서/JKB O
승리/NNG O
를/JKO O
얻/VV O
었/EP O
다/EC O
./SF O
```

3. Pretrained embedding을 이용한 실험 결과

i. 숙제 2

[ train.py ]

```
Epoch 1 out of 10
acc : 97.11 - f1 : 22.08
new best score
Epoch 2 out of 10
acc : 97.30 - f1 : 38.38
new best score
Epoch 3 out of 10
acc : 97.57 - f1 : 45.36
new best score
Epoch 4 out of 10
acc : 97.55 - f1 : 52.60
new best score
Epoch 5 out of 10
acc : 97.49 - f1 : 52.95
new best score
Epoch 6 out of 10
acc : 97.40 - f1 : 53.51
new best score
Epoch 7 out of 10
acc : 97.56 - f1 : 52.71
Epoch 8 out of 10
acc : 97.48 - f1 : 49.92
Epoch 9 out of 10
acc : 97.40 - f1 : 52.70
- early stopping 3 epochs without improvement
```

[ evaluate.py ]

```
Reloading the latest trained model...
Testing model over test set
acc : 97.26 - f1 : 48.36
23/SN B-DT
일/NNB I-DT
기성용/NNP O
의/JKG O
활약/NNG O
으로/JKB O
스완지시티/NNP O
는/JX O
리버풀/NNP O
선/NNG O
에서/JKB O
승리/NNG O
를/JKO O
얻/VV O
었/EP O
다/EC O
./SF O
```

ii. 숙제 3

[ train.py ]

```
Epoch 1 out of 10
acc : 96.98 - f1 : 17.43
new best score
Epoch 2 out of 10
acc : 97.43 - f1 : 44.25
new best score
Epoch 3 out of 10
acc : 97.46 - f1 : 48.90
new best score
Epoch 4 out of 10
acc : 97.47 - f1 : 53.86
new best score
Epoch 5 out of 10
acc : 97.67 - f1 : 56.94
new best score
Epoch 6 out of 10
acc : 97.66 - f1 : 55.75
Epoch 7 out of 10
acc : 97.66 - f1 : 53.70
Epoch 8 out of 10
acc : 97.50 - f1 : 49.43
- early stopping 3 epochs without improvement
```

[ evaluate.py ]

```
Reloading the latest trained model...
Testing model over test set
acc : 97.59 - f1 : 54.01
23/SN B-DT
일/NNB I-DT
기성용/NNP O
의/JKG O
활약/NNG O
으로/JKB O
스완지시티/NNP O
는/JX O
리버풀/NNP O
선/NNG O
에서/JKB O
승리/NNG O
를/JKO O
얻/VV O
었/EP O
다/EC O
./SF O
```