



BERT NER

최종 발표



NLP209



프로젝트 주제

주제: BERT를 이용한 키워드 추출

목적1) BERT모델을 이용하여 BIO tagging data로 fine-tuning을 하여 NER task를

수행하도록 해 봄으로써 모델 이해

2) 기존의 biobert는 tensorflow로 이루어져 있어 코드를 이해하는데 어려움 多

→ 우리는 이를 keras로 수정하여

최종 발표

사용한 Data: BC5CDR-disease의 train, validation, test data

train - Windows 메모장		
파일(F)	편집(E)	서식(O) 보기(V)
degree	O	
of	O	
orthostatic		B
hypotension		I
occurred	O	
with	O	
standing	O	
.	O	
Orthostatic		B
hypotension		I
was	O	

train_dev - Windows 메모장		
파일(F)	편집(E)	서식(O) 보기(V) 도
Selegiline	O	
-	O	
induced	O	
postural	B	
hypotension		I
in	O	
Parkinson	B	
'	I	
s	I	
disease	I	
:	O	
a	O	
longitudinal		O
study	O	
on	O	

test - Windows 메모장		
파일(F)	편집(E)	서식(O) 보기(V)
Torsade	B	
de	I	
pointes	I	
ventricular		B
tachycardia		I
during	O	
low	O	
dose	O	
intermittent		O
dobutamine		O
treatment	O	
in	O	

최종 발표

이틀간 진행한 내용

1. data를 읽어와 단어의 총 개수가 30을 넘지 않도록 data 자르기 (B, I 사이에 잘리지 않게) → [labels, sentence] 형식의 examples 생성

```
[['0000B10B1110000000000000',
  "Selegiline - induced postural hypotension in Parkinson ' s disease : a longitudinal study on the effects of drug withdrawal ."],
 ['000000B1110000000000000000B1110B00',
  "OBJECTIVES : The United Kingdom Parkinson ' s Disease Research Group ( UKPDRG ) trial found an increased mortality in patients with Parkins",
 ['000000000000000000000000',
  'to receive 10 mg selegiline per day and L - dopa compared with those taking L - dopa alone .'],
 ['000000000000000000000000B110000000000']]
```

```
import csv
df_en=pd.read_table(path+"/train.tsv",delimiter='\\t',quoting=csv.QUOTE_NONE)
```

```
def _read_data(input_file):
    with open(input_file) as f:
        lines = []
        words = []
        labels = []
        for line in f:
            contends = line.strip()
            if len(contends) == 0:
                assert len(words) == len(labels)
                if len(words) > 30:
                    while len(words) > 30:
                        tlabel = labels[:30]
                        for tidx in range(len(tlabel)):
                            if tlabel.pop() == '0':
                                break
                        l = ' '.join([label for label in labels[:len(tlabel)+1] if len(label) > 0])
                        w = ' '.join([word for word in words[:len(tlabel)+1] if len(word) > 0])
                        lines.append([l,w])
                        words = words[len(tlabel)+1:]
                        labels = labels[len(tlabel)+1:]
                if len(words) == 0:
                    continue
                l = ' '.join([label for label in labels if len(label) > 0])
                w = ' '.join([word for word in words if len(word) > 0])
                lines.append([l, w])
                words = []
                labels = []
                continue
            word = line.strip().split()[0]
            label = line.strip().split()[-1]
            words.append(word)
            labels.append(label)
        return lines
```

데이터를 table로 불러올때

최종 발표

이틀간 진행한 내용

- data(BC5CDR-disease)를 input data, target data 형식으로 변환

- 2. 각각의 example들을 WordPieceTokenizer로 토큰화

- biobert의 Tokenization을 우리 코드 상에서 구현.

- 이를 이용하여 tokenize 진행

LICENSE	initial
README.md	Update README.md
init.py	initial
create_pretraining_data.py	initial
download.sh	download script
extract_features.py	initial
modeling.py	initial
modeling_test.py	initial
optimization.py	initial
optimization_test.py	initial
requirements.txt	fix requirements
run_classifier.py	initial
run_ner.py	initial
run_pretraining.py	initial
run_qa.py	initial
run_re.py	initial
sample_text.txt	initial
tf_metrics.py	initial
tokenization.py	initial
tokenization_test.py	initial

Biobert의
tokenization.py를
우리 코드에 올릴 수 있도록
run_ner.py 코드 수정

```
text = convert_to_unicode(text)
```

```
output_tokens = []
for token in whitespace_tokenize(text):
    chars = list(token)
    if len(chars) > self.max_input_chars_per_word:
        output_tokens.append(self.unk_token)
        continue

    is_bad = False
    start = 0
    sub_tokens = []
    while start < len(chars):
        end = len(chars)
        cur_substr = None
        while start < end:
            substr = "".join(chars[start:end])
            if start > 0:
                substr = "##" + substr
            if substr in self.vocab:
                cur_substr = substr
                break
            end -= 1
        if cur_substr is None:
            is_bad = True
            break
        sub_tokens.append(cur_substr)
        start = end

    if is_bad:
        output_tokens.append(self.unk_token)
    else:
        output_tokens.extend(sub_tokens)
return output_tokens
```

Tokenization의 Wordpiecetokenizer 부분

최종 발표

이틀간 진행한 내용

- data(BC5CDR-disease)를 input data, target data 형식으로 변환

3. convert_single_example 함수를 이용하여 input_ids, segment_ids, label_ids

- input_ids, segment_ids : train_x

- label_ids: train_y


```

def convert_single_example(ex_index, example, label_list, max_seq_length, tokenizer):
    #vocab=path+'/vocab.txt' #구드path로 변경!
    # wordtt= FullTokenizer(vocab) #class형태로 선언

    label_map = {}
    for (i, label) in enumerate(label_list,1):
        label_map[label] = i
    #label_map: {'B': 1, 'I': 2, 'O': 3, 'X': 4, '[CLS]': 5, '[SEP]': 6}
    # print("label_map: ", label_map)
    # with open(os.path.join(FLAGS.output_dir, 'label2id.pkl'), 'wb') as w:
    #     pickle.dump(label_map, w)

    textlist = example[1].split(' ') #example.text.split(' ')을 index형태로 바꿈
    labellist = example[0].split(' ') #example.label.split(' ')을 index형태로 바꿈
    tokens = []
    labels = []

    for i, word in enumerate(textlist):
        token = tokenizer.tokenize(word) #사용!!
        tokens.extend(token)
        label_1 = labellist[i]
        for m in range(len(token)):
            if m==0:
                labels.append(label_1)
            else:
                labels.append("X")

    if len(tokens) >= max_seq_length - 1:
        tokens = tokens[0:(max_seq_length - 2)]
        labels = labels[0:(max_seq_length - 2)]

    ntokens = []
    segment_ids = []
    label_ids = []
    ntokens.append("[CLS]")
    segment_ids.append(0)
    # append("0") or append("[CLS]") not sure!
    label_ids.append(label_map["[CLS]"])
    for i, token in enumerate(tokens):
        ntokens.append(token)

```

- Mask 변수 삭제

- [PAD]: 0

- 객체 사용 X, 인덱스 사용

- “케라스” → B/I/O 中 B

→ “케라스”를 tokenization하면 [케, ##라, ##스]

→ labels=[B, X, X]

```
vocab=path+ '/vocab.txt' → BioBERT의 vocab  
label_list=['B','I','O','X','[CLS]','[SEP]'] #101: [CLS], 102: [SEP]  
max_seq_length= 128 → max_seq_length  
tokenizer= FullTokenizer(vocab) → tokenizer
```

```
result_input_ids=[]  
result_seg_ids=[]  
y=[]
```

```
for i, example in enumerate(ex):  
    input_ids, segment_ids, label_ids=convert_single_example(i, example, label_list,max_seq_length, tokenizer)  
    result_input_ids.append(input_ids)  
    result_seg_ids.append(segment_ids)  
  
    y.append(label_ids)  
  
train_x=[]  
train_y=[]  
train_x.append(np.array(result_input_ids))  
train_x.append(np.array(result_seg_ids))  
  
train_y.append(np.array(y))
```

최종 발표

train_x data

```
[array([[ 101, 14516, 27412, ..., 0, 0, 0],
       [ 101, 11350,  131, ..., 0, 0, 0],
       [ 101,  1106, 3531, ..., 0, 0, 0],
       ...,
       [ 101,  175, 1377, ..., 0, 0, 0],
       [ 101,  1103, 3469, ..., 0, 0, 0],
       [ 101,  1292, 2686, ..., 0, 0, 0]]),
 array([[0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       ...,
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0]])]
```

train_y data

```
[array([[5, 3, 4, ..., 0, 0, 0],
       [5, 3, 3, ..., 0, 0, 0],
       [5, 3, 3, ..., 0, 0, 0],
       ...,
       [5, 3, 4, ..., 0, 0, 0],
       [5, 3, 3, ..., 0, 0, 0],
       [5, 3, 3, ..., 0, 0, 0]])]
```

{'B': 1, 'I': 2, 'O': 3, 'X': 4, '[CLS]': 5, '[SEP]': 6}

```
SEQ_LEN = 128  
BATCH_SIZE = 16  
EPOCHS=2  
LR=1e-5
```

```
pretrained_path = "gdrive/My Drive/HUB/biobert/biobert_v1.1_pubmed"
```

```
os.listdir(pretrained_path)
```

```
['.ipynb_checkpoints',  
 'bert_config.json',  
 'model.ckpt-1000000.index',  
 'model.ckpt-1000000.meta',  
 'vocab.txt',  
 'model.ckpt-1000000.data-00000-of-00001']
```

```
config_path = os.path.join(pretrained_path, 'bert_config.json')  
checkpoint_path = os.path.join(pretrained_path, 'model.ckpt-1000000')  
vocab_path = os.path.join(pretrained_path, 'vocab.txt')
```

```
layer_num = 12  
model = load_trained_model_from_checkpoint( #사전학습된 모델 불러오기  
    config_path,  
    checkpoint_path,  
    training=False,  
    trainable=True,  
    seq_len=SEQ_LEN)
```

Biobert의 weight를 사용하기 위해

최종 발표

Bert 모형의 자체 Masking 된 텐서들을
풀어주기 위한 NonMasking

```
from keras.layers import Layer
class NonMasking(Layer):
    def __init__(self, **kwargs):
        self.supports_masking = True
        super(NonMasking, self).__init__(**kwargs)

    def build(self, input_shape):
        input_shape = input_shape

    def compute_mask(self, input, input_mask=None):
        return None

    def call(self, x, mask=None):
        return x

    def get_output_shape_for(self, input_shape):
        return input_shape
```

```
class MyLayer(Layer):
    def __init__(self, seq_len, **kwargs):

        self.seq_len = seq_len
        self.supports_masking = True
        super(MyLayer, self).__init__(**kwargs)
```

```
def build(self, input_shape):각 token별로 output값이 layer 개수만큼 나오게 하기 위해
    self.W = self.add_weight(name='kernel', shape=(768, 7), initializer='uniform', trainable=True)
    super(MyLayer, self).build(input_shape)
```

```
def call(self, x):
    x = K.reshape(x, shape=(-1, 128, 768)) Shape 를 max_seq에 맞게
    x = K.dot(x, self.W)
```

```
# Ner은 각 token별로 softmax를 취해서 label을 유추하므로 이 부분은 제외
# x = K.permute_dimensions(x, (2, 0, 1))
# self.start_logits, self.end_logits = x[ 0], x[1]
#x = K.reshape(x, shape=(-1, 128, 7))
```

```
self.logits = K.softmax(x, axis= -1) Token별로 softmax
```

```
#self.logits = K.reshape(logit, shape=(-1, 128*7))
```

```
return self.logits
```

```
def compute_output_shape(self, input_shape):
    return (input_shape[0], self.seq_len)
```

최종 발표

```
def get_bert_finetuning_model(model):  
    inputs = model.inputs[:2] #segment, token 두개의 input  
    dense = model.output  
    x = NonMasking()(dense)  
    output_layer = MyLayer(128)(x)  
    model = keras.models.Model(inputs, output_layer)  
    model.compile(  
        optimizer=Adam(learning_rate=LR, weight_decay=0.001),  
        loss="sparse_categorical_crossentropy",  
        metrics=["accuracy"])  
    #loss='categorical_crossentropy',  
    #metrics=['categorical_accuracy'])  
  
    return model
```

우리의 layer를 쓰기 위한 함수

최종 발표

```
sess = K.get_session()
uninitialized_variables = set([i.decode('ascii') for i in sess.run(tf.report_uninitialized_variables())])
init = tf.variables_initializer([v for v in tf.global_variables() if v.name.split(':')[0] in uninitialized_variables])
sess.run(init)

bert_model = get_bert_finetuning_model(model)
bert_model.summary()
# bert_model.fit 함수에 들어가긴 하는데, error가 남
train_y_hot_squ = np.reshape(train_y_hot, (-1,128*7))
history = bert_model.fit(train_x, train_y_hot_squ, batch_size=10, validation_split=0.05, shuffle=False, verbose=1)
```


Model summary

Encoder-12-MultiHeadSelfAttention	(None, 128, 768)	2362368	Encoder-11-FeedForward-Norm[0][0]
Encoder-12-MultiHeadSelfAttention	(None, 128, 768)	0	Encoder-12-MultiHeadSelfAttention
Encoder-12-MultiHeadSelfAttention	(None, 128, 768)	0	Encoder-11-FeedForward-Norm[0][0] Encoder-12-MultiHeadSelfAttention
Encoder-12-MultiHeadSelfAttention	(None, 128, 768)	1536	Encoder-12-MultiHeadSelfAttention
Encoder-12-FeedForward (FeedFor	(None, 128, 768)	4722432	Encoder-12-MultiHeadSelfAttention
Encoder-12-FeedForward-Dropout	(None, 128, 768)	0	Encoder-12-FeedForward[0][0]
Encoder-12-FeedForward-Add (Add	(None, 128, 768)	0	Encoder-12-MultiHeadSelfAttention Encoder-12-FeedForward-Dropout [0]
Encoder-12-FeedForward-Norm (La	(None, 128, 768)	1536	Encoder-12-FeedForward-Add[0][0]
non_masking_16 (NonMasking)	(None, 128, 768)	0	Encoder-12-FeedForward-Norm[0][0]
my_layer_16 (MyLayer)	(None, 128, 768)	5376	non_masking_16[0][0]

Total params: 107,430,144
Trainable params: 107,430,144
Non-trainable params: 0

Data flow

Torsade de pointes ventricular tachycardia during low.

B, I, O tagging된 data

↓ read_data

['B I I ... O', 'Torsade de ...']

↓ convert_single_example

input_ids, segment_ids, label_ids

→ train_x

→ train_y

len
= max_seq

input_ids = [각 단어의 vocabulari 인덱스]

segment_ids = [~~첫번째~~ / 두번째 문장인지 → 0 / 1] ⇒ 다 0

label_ids = [1, 2, 2, ...]

{ 'B': 1, 'I': 2, 'O': 3, 'X': 4, '[CLS]': 5, '[SEP]': 6 }

train_x, train_y

↓ model

output
(128, 7)

학습 결과

Model.fit 과정에서 문제가 있어, 아직 확인 못함!

프로젝트 진행 중 겪은 문제와 해결

1. biobert pre-trained weight를 colab에 올리는 과정



```
config_path = os.path.join(pretrained_path, 'bert_config.json')
checkpoint_path = os.path.join(pretrained_path, 'model.ckpt-1000000')
vocab_path = os.path.join(pretrained_path, 'vocab.txt')
```

```
layer_num = 12
```

```
model = load_trained_model_from_checkpoint( #사전 학습된 모델 불러오기
```

```
    config_path,
    checkpoint_path,
    training=True,
    trainable=True,
    seq_len=SEQ_LEN,)
```

DataLossError Traceback (most recent call last)

<ipython-input-24-5a585d21b790> in <module>()

```
12     training=True,
13     trainable=True,
--> 14     seq_len=SEQ_LEN,)
```

6 frames

/usr/local/lib/python3.6/dist-packages/tensorflow_core/python/pywrap_tensorflow_internal.py in __init__(self, file

```
883
884     def __init__(self, filename):
--> 885         this = _pywrap_tensorflow_internal.new_CheckpointReader(filename)
886         try:
887             self.this.append(this)
```

DataLossError: block checksum mismatch

프로젝트 진행 중 겪은 문제와 해결

1. biobert pre-trained weight를 colab에 올리는 과정

* 원인

: 이미 이전에 pre-trained weight로 fine-tuning을 한번 진행했었는데,
그 weight를 다시 사용하니 checkpoint에 충돌 발생

* 해결

: bioBert의 weight를 다시 새롭게 다운받아서 사용

프로젝트 진행 중 겪은 문제와 해결

2. Model.fit을 하는 과정에서

```
4 frames
/usr/local/lib/python3.6/dist-packages/tensorflow_core/python/client/session.py in __call__(self,
1470     ret = tf_session.TF_SessionRunCallable(self._session._session,
1471                                             self._handle, args,
-> 1472                                             run_metadata_ptr)
1473     if run_metadata:
1474         proto_data = tf_session.TF_GetBuffer(run_metadata_ptr)
```

InvalidArgumentError: 2 root error(s) found.

(0) Invalid argument: Incompatible shapes: [10] vs. [10,128]

[[{{(node metrics_5/acc/Equal)}}]]

[[loss_5/mul/_43325]]

(1) Invalid argument: Incompatible shapes: [10]

[[{{(node metrics_5/acc/Equal)}}]]

0 successful operations.

0 derived errors ignored.

ValueError

Traceback (most recent call last)

<ipython-input-133-4d28e6cb8ef6> in <module>()

7 bert_model.summary()

8 # bert_model.fit 함수에 들어가긴 하는데, error가 남

----> 9 history = bert_model.fit(train_x, train_y, batch_size=10, validation_split=0.05, shuffle=False, verbose=1)

```
2 frames
/usr/local/lib/python3.6/dist-packages/keras/engine/training_utils.py in standardize_input_data(data, names, shapes, check_batch_axis, exception_prefix)
139     ': expected ' + names[i] + ' to have shape ' +
140     str(shape) + ' but got array with shape ' +
-> 141     str(data_shape))
142     return data
143
```

ValueError: Error when checking target: expected my_layer_9 to have shape (1,) but got array with shape (128,)

프로젝트 진행 중 겪은 문제와 해결

2. Model.fit을 하는 과정에서

- 원인

: cross-entropy할때



→ train_y



→ output

프로젝트 진행 중 겪은 문제와 해결

2. Model.fit을 하는 과정에서 → 시도 1 train_y의 모든 vector를 one-hot vector로

→ train_y의 각 Token 별 one-hot vector



→ train_y의 one-hot vector



→ output

프로젝트 진행 중 겪은 문제와 해결

2. Model.fit을 하는 과정에서

실패한 원인

: 이전 - loss를 token 별로 X, input 문장 별로 구했음

: NER- 문장을 구성하는 token별로 구해야 해서

→ Cross entropy를 사용하지 않고, loss/ matrix에 대한 class를 따로 지정해서 한번 더 해볼 예정

최종 결과 알고리즘 코드

<https://drive.google.com/open?id=1dC5csukxyu33w3SvQulHpADI3YfIMIDT>