

# bilstm-crf命名实体识别

## 1、环境

- python3.6
- jieba=0.37
- Tensorflow=1.2.0

## 2、模型文件

- 其中 ckpt , ckpt\_book , log , maps , result , result\_book , maps.pkl , maps\_book.pkl , train.log , train\_book.log 等文件，是训练模型时产生的文件。

### (1) data\_processing.py文件

用于语料库的处理，全部处理成小于max\_seq\_length的序列，这样可以避免解码出现不合法的数据或者在最后算结果的时候出现out of range 的错误。

主要有句子的切断，加载文件和输出文件的方法。

### (2) data\_utils.py文件

该文件主要是创建字典，字符之间的转换，语料标注格式的检测，加载词向量，以及数据batch化。

### (3) loader.py文件

针对准备好的领域语料，调用data\_utils.py文件中的方法，将语料处理成模型可使用的数据。

### (4) main.py文件

该模型主要是参数设置，模型的训练和测试。

### (5) utils.py

该文件是模型的加载，存储，以及加载存储路径的设置。

## 3、模型训练

其训练的代码和参数如下面所示：

```
1 con_key_book = {
2     'train_type': False,
3     'ckpt_path': 'ckpt_book',
4     'summary_path': 'summary_book',
5     'log_file': 'train_book.log',
6     'map_file': 'maps_book.pkl',
```

```

7  'result_path': 'result_book',
8  'train_file': 'train_book.data',
9  'test_file': 'test_book.data',
10 'dev_file': 'dev_book.data',
11 }
12 ner_model = NER_LSTM_CRF(con_key_book)
13 if ner_model.FLAGS.train:
14     if ner_model.FLAGS.clean:
15         clean(ner_model.FLAGS)
16     ner_model.train()
17 else:
18     ner_model.evaluate_line()

```

其中需要模型参数的设置，已经训练结果。

## 4、模型测试

其中需要输入参数，已经测试的数据，其测试代码如下：

```

1  def test(con_key,sentence):
2      ner_model = NER_LSTM_CRF(con_key)
3      config = load_config(ner_model.FLAGS.config_file)
4      logger = get_logger(ner_model.FLAGS.log_file)
5      # limit GPU memory
6      tf_config = tf.ConfigProto()
7      tf_config.gpu_options.allow_growth = True
8      with open(ner_model.FLAGS.map_file, "rb") as f:
9          char_to_id, id_to_char, tag_to_id, id_to_tag = pickle.load(f)
10         result_list = []
11         with tf.Session(config=tf_config) as sess:
12             model = create_model(sess, Model, ner_model.FLAGS.ckpt_path, load_word2
vec, config, id_to_char, logger)
13             for line in sentence:
14                 result = model.evaluate_line(sess, input_from_line(line, char_to_id), i
d_to_tag)
15                 result_list.append(result)
16         return result_list

```

其调用上面函数的代码为：

```

1  con_key_qa = {
2      'trian_type': False,
3      'ckpt_path': 'ckpt',
4      'summary_path': 'summary',
5      'log_file': 'train.log',

```

```
6  'map_file': 'maps.pkl',
7  'result_path': 'result',
8  'train_file': 'train.data',
9  'test_file': 'test.data',
10 'dev_file': 'dev.data',
11 }
12 sentence = ['小规模纳税人为什么可以底销项税不能底进项税']
13 # print('book',test(con_key_book, sentence))
14 print('qa',test(con_key_qa,sentence))
```

基于上面的代码既可以完成测试，也可以部署到工程应用中。