用 BERT 进行命名实体识别过程的报告

---- 以人民日报文本信息为例

李明聪

摘要：本文用标注后的人民日报信息作为数据集，使用 BERT 进行了命名实体识别。数据集共 240 万个字符，模型在 GPU 上训练耗时约 50 分钟，模型预测的 f1 score 为 0.96。

一、数据集概述

1. 数据来源：人民日报 1998 年上半年的文本语料，进行了分词和词标注。

2. 数据集说明：

训练集共 222 万个字符，验证集共 18 万个字符，测试集共 2362 个字符。

2. 标注说明

(1) 实体位置的标注

标注	含义	含义
B-X	Begin	代表实体 X 的开头
I-X	Inside	代表实体的内部
O-X	outside	代表不属于任何类型的

(2) 实体类型的标注：X

LOC	地名
PER	人民
ORG	机构名

4. 示例：

《	北	京	文	物	保	存	保	管	状	态	之	调	查	报	告	》
O	B-LOC	I-LOC	O	O	O	O	O	O	O	O	O	O	O	O	O	O
调	查	范	围	涉	及	故	宫	、	历	博	、	古	研	所		
O	O	O	O	O	O	B-LOC	I-LOC	O	B-LOC	I-LOC	O	B-ORG	I-ORG	I-ORG		

二、工作环境

Google CoLab

三、训练过程

1. 准备好所有用到的文件。

* bert：从 BERT 官方 GitHub 仓库下载的代码

* checkpoint：存放中文预训练模型

* data：要用到的数据集

* BERT_NER：运行这个 py 文件进行学习与预测。

bert	2020/9/20 23:26
checkpoint	2020/9/20 23:26
data	2020/9/20 23:26
output	2020/9/21 15:46
Analysis Report.docx	2020/9/25 12:47
BERT_NER.py	2020/9/20 19:44

2. 训练模型

```
!python BERT_NER.py \
                --data_dir=data/ \
                --bert_config_file=checkpoint/bert_config.json \
                --init_checkpoint=checkpoint/bert_model.ckpt \
                --vocab_file=vocab.txt \
                --output_dir=./output/result_dir/
```

在 CoLab 里面可以通过 Linux 命令运行 py 文件。

除第一行外，其他行均为传入的参数。

训练完毕的截图：

```
I0920 14:41:37.346732 140295888770944 tpu_estimator.py:2307] global_step/sec: 1.11866
INFO:tensorflow:examples/sec: 35.7972
I0920 14:41:37.347095 140295888770944 tpu_estimator.py:2308] examples/sec: 35.7972
INFO:tensorflow:global_step/sec: 1.11456
I0920 14:41:38.243934 140295888770944 tpu_estimator.py:2307] global_step/sec: 1.11456
INFO:tensorflow:examples/sec: 35.6658
I0920 14:41:38.244330 140295888770944 tpu_estimator.py:2308] examples/sec: 35.6658
INFO:tensorflow:Saving checkpoints for 4749 into ./output/result_dir/model.ckpt.
I0920 14:41:38.245431 140295888770944 basic_session_run_hooks.py:606] Saving checkpoints for 4749
WARNING:tensorflow:From /tensorflow-1.15.2/python3.6/tensorflow_core/python/training/saver.py:963:
Instructions for updating:
Use standard file APIs to delete files with this prefix.
W0920 14:41:42.877460 140295888770944 deprecation.py:323] From /tensorflow-1.15.2/python3.6/tensor
Instructions for updating:
Use standard file APIs to delete files with this prefix.
INFO:tensorflow:Loss for final step: 2.2763486.
I0920 14:41:44.157709 140295888770944 estimator.py:371] Loss for final step: 2.2763486.
INFO:tensorflow:training_loop marked as finished
I0920 14:41:44.158624 140295888770944 error_handling.py:101] training_loop marked as finished
```

3. 预测

相比训练模型时，增加三个参数，设置 do_predict=True，do_train=False.

```
!python BERT_NER.py \
                --data_dir=data/ \
                --bert_config_file=checkpoint/bert_config.json \
                --init_checkpoint=checkpoint/bert_model.ckpt \
                --vocab_file=vocab.txt \
                --output_dir=./output/result_dir/ \
                --do_train=False \
                --do_eval=True \
                --do_predict=True
```

预测的完成的截图。

```
name: Tesla T4 major: 7 minor: 5 memoryClockRate(GHz): 1.59
pciBusID: 0000:00:04.0
2020-09-20 14:49:58.203643: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successf
2020-09-20 14:49:58.203684: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successf
2020-09-20 14:49:58.203709: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successf
2020-09-20 14:49:58.203734: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successf
2020-09-20 14:49:58.203756: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successf
2020-09-20 14:49:58.203773: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successf
2020-09-20 14:49:58.203794: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successf
2020-09-20 14:49:58.203883: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:983] successful N
2020-09-20 14:49:58.204441: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:983] successful N
2020-09-20 14:49:58.204922: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1767] Adding visible
2020-09-20 14:49:58.204981: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1180] Device intercon
2020-09-20 14:49:58.204998: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1186]      0
2020-09-20 14:49:58.205008: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1199] 0:   N
2020-09-20 14:49:58.205108: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:983] successful N
2020-09-20 14:49:58.205695: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:983] successful N
2020-09-20 14:49:58.206248: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1325] Created TensorF
INFO:tensorflow:Restoring parameters from ./output/result_dir/model.ckpt-4749
I0920 14:49:58.208252 139852449249152 saver.py:1284] Restoring parameters from ./output/result_dir/m
INFO:tensorflow:Running local_init_op.
I0920 14:49:59.342778 139852449249152 session_manager.py:500] Running local_init_op.
INFO:tensorflow:Done running local_init_op.
I0920 14:49:59.406609 139852449249152 session_manager.py:502] Done running local_init_op.
INFO:tensorflow:prediction_loop marked as finished
I0920 14:50:00.726061 139852449249152 error_handling.py:101] prediction_loop marked as finished
INFO:tensorflow:prediction_loop marked as finished
I0920 14:50:00.726294 139852449249152 error_handling.py:101] prediction_loop marked as finished
```

预测返回的结果：

token_test.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V

```
[CLS]
美
国
的
华
莱
士
，
我
和
他
谈
笑
风
生
。
[SEP]
[CLS]
看
包
```

测试集的 token

label_test.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(

```
[CLS]
B-LOC
I-LOC
O
B-LOC
I-LOC
I-LOC
O
O
O
O
O
O
O
O
O
[SEP]
[CLS]
O
B-PER
O
```

预测的 label

自动生成常见的评价指标。

eval_results.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
eval_f = 0.96219593
eval_precision = 0.96075124
eval_recall = 0.9638879
global_step = 4749
loss = 15.491793
```