一个开箱即用的BERT_BiLSTM_CRF实体识别工具

原创 西兰 自然语言处理与算法 2020-12-03

收录于话题
#NLP实战工具
6个

仓库地址：https://github.com/StanleyLsx/entity_extractor_by_ner

实体识别

此仓库是基于Tensorflow2.3的NER任务项目，既可以使用BiLSTM-Crf模型，也可以使用Bert-BiLSTM-Crf模型，提供可配置文档，**配置完可直接运行。**

环境

- **CPU**: tensorflow==2.3.0
- **GPU**: tensorflow-gpu==2.3.0
- tensorflow-addons==0.11.2
- transformers==3.0.2
- python 3.6.7

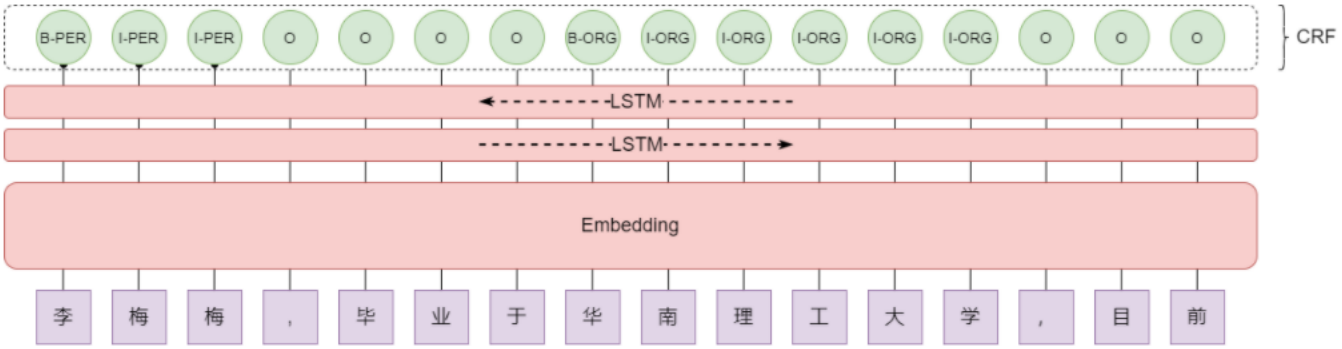
集群下推荐GPU加速训练，其他环境见requirements.txt

数据集

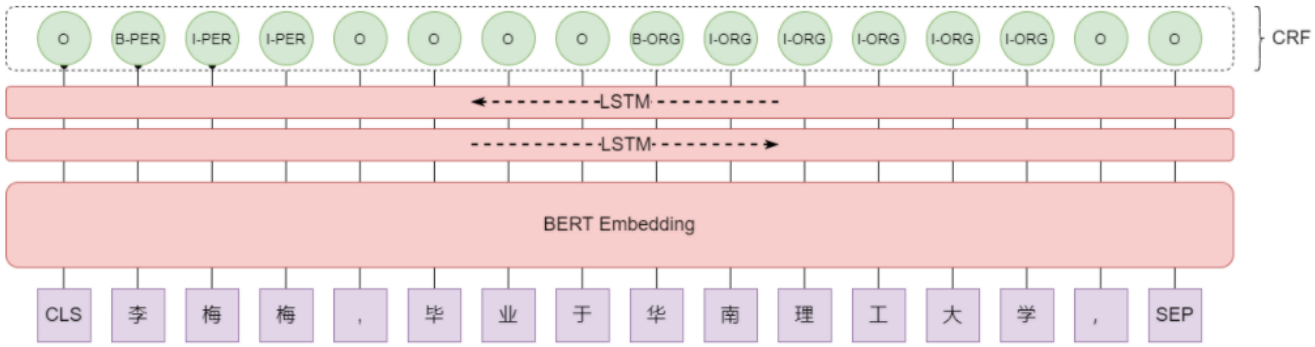
人民日报语料

原理

Bilstm-CRF



Bert-Bilstm-CRF



CRF层

最通俗易懂的BiLSTM-CRF模型中的CRF层介绍
CRF Layer on the Top of BiLSTM - 1
CRF层需要使用viterbi译码法，知乎上这个答案比较容易理解

使用

训练

将已经标注好的数据切割好训练、验证集放入data目录下，如果只提供训练集将会有程序自己按照9:1切割训练集与验证集。
在system.config的Datasets(Input/Output)下配置好数据集的路径、分隔符、模型保存地址等。
在system.config的Labeling Scheme配置标注模式。
在system.config的Model Configuration/Training Settings下配置模型参数和训练参数。

设定system.config的Status中的为train:

```
################ Status ################
mode=train
# string: train/interactive_predict
```

是否使用Bert做embedding(选择True/False):

```
use_bert=False
```

运行main.py开始训练。

- Bilstm-CRF模型下效果

```
training batch:   520, loss: 0.60883, precision: 0.955 recall: 0.928 f1: 0.941 accuracy: 0.991
training batch:   540, loss: 0.39766, precision: 0.941 recall: 0.914 f1: 0.928 accuracy: 0.992
training batch:   560, loss: 0.17681, precision: 0.978 recall: 0.978 f1: 0.978 accuracy: 0.999
training batch:   580, loss: 0.52320, precision: 0.908 recall: 0.967 f1: 0.937 accuracy: 0.985
training batch:   600, loss: 0.43353, precision: 0.983 recall: 0.967 f1: 0.975 accuracy: 0.994
training batch:   620, loss: 0.92548, precision: 0.981 recall: 0.930 f1: 0.955 accuracy: 0.992
training batch:   640, loss: 0.13315, precision: 1.000 recall: 1.000 f1: 1.000 accuracy: 1.000
training batch:   660, loss: 0.21323, precision: 1.000 recall: 0.956 f1: 0.977 accuracy: 0.997
training batch:   680, loss: 0.92134, precision: 0.930 recall: 0.964 f1: 0.946 accuracy: 0.980
training batch:   700, loss: 2.06022, precision: 0.972 recall: 0.814 f1: 0.886 accuracy: 0.966
training batch:   720, loss: 0.50172, precision: 0.955 recall: 0.928 f1: 0.941 accuracy: 0.993
start evaluate engines...
label: ORG, precision: 0.828 recall: 0.753 f1: 0.781 accuracy: 0.000
label: PER, precision: 0.930 recall: 0.850 f1: 0.881 accuracy: 0.000
label: LOC, precision: 0.856 recall: 0.871 f1: 0.858 accuracy: 0.000
time consumption:6.76(min), precision: 0.889 recall: 0.849 f1: 0.867 accuracy: 0.981
saved the new best model with f1: 0.867
```

- Bert-Blism-CRF模型下效果

```
training batch:   480, loss: 0.50103, precision: 0.923 recall: 0.968 f1: 0.945 accuracy: 0.989
training batch:   500, loss: 0.02047, precision: 1.000 recall: 1.000 f1: 1.000 accuracy: 1.000
training batch:   520, loss: 0.26341, precision: 0.923 recall: 0.923 f1: 0.923 accuracy: 0.993
training batch:   540, loss: 0.22349, precision: 1.000 recall: 0.978 f1: 0.989 accuracy: 0.995
training batch:   560, loss: 0.30378, precision: 0.948 recall: 0.965 f1: 0.957 accuracy: 0.996
training batch:   580, loss: 0.14488, precision: 0.977 recall: 0.977 f1: 0.977 accuracy: 0.999
training batch:   600, loss: 0.40148, precision: 0.958 recall: 0.939 f1: 0.948 accuracy: 0.996
training batch:   620, loss: 0.22603, precision: 1.000 recall: 0.966 f1: 0.982 accuracy: 0.999
training batch:   640, loss: 0.33313, precision: 0.983 recall: 0.983 f1: 0.983 accuracy: 0.995
training batch:   660, loss: 0.21998, precision: 1.000 recall: 0.943 f1: 0.971 accuracy: 0.998
training batch:   680, loss: 0.17948, precision: 0.986 recall: 0.986 f1: 0.986 accuracy: 0.995
training batch:   700, loss: 0.32754, precision: 0.968 recall: 0.968 f1: 0.968 accuracy: 0.996
training batch:   720, loss: 0.37428, precision: 0.962 recall: 0.926 f1: 0.943 accuracy: 0.993
start evaluate engines...
label: ORG, precision: 0.898 recall: 0.882 f1: 0.886 accuracy: 0.000
label: PER, precision: 0.968 recall: 0.975 f1: 0.968 accuracy: 0.000
label: LOC, precision: 0.931 recall: 0.932 f1: 0.929 accuracy: 0.000
time consumption:82.86(min), precision: 0.948 recall: 0.945 f1: 0.946 accuracy: 0.993
saved the new best model with f1: 0.946
```

注(1):这里使用的transformers包加载Bert，初次使用的时候会自动下载Bert的模型

注(2):当重新训练的时候，Bert-Bilstm-CRF和Bilstm-CRF各自自动生成自己vocabs/label2id文件，不能混用，如果需要共用，你可以手动的定义标签
注(3):使用Bert-Bilstm-CRF时候max_sequence_length不能超过512并且embedding_dim默认为768

在线预测

仓库中已经训练好了两种模型在同一份数据集上的参数可直接进行试验，两者位于data/example_datasets目录下

- 使用Bilstm-CRF模型时使用system.config4bilstm-crf的配置
- 使用Bert-Bilstm-CRF模型时使用system.config4bert-bilstm-crf的配置
  将对应的配置命名为system.config然后替换掉当前的配置。

如果重新训练，务必保留system.config文件，设定system.config的Status中的为interactive_predict。

```
################ Status ################
mode=interactive_predict
# string: train/interactive_predict
```

最后，运行main.py开始在线预测。
下图为在线预测结果，你可以移植到自己项目里面做成对外接口。

```
loading model successfully
please input a sentence (enter [exit] to exit.)
18还是19年江小白不是有广州站么
18还是19年江小白不是有广州站么
2020-09-13 07:36:30.403009: W tensorflow/core/grappler/optimizers/loop_optimizer.cc:9
(['江 小 白', '广 州'], ['PER', 'LOC'], [(7, 10), (13, 15)])
please input a sentence (enter [exit] to exit.)
南澳风景很好，环岛骑过。青澳湾风景最好
南澳风景很好，环岛骑过。青澳湾风景最好
(['南 澳', '青 澳 湾'], ['LOC', 'LOC'], [(0, 2), (12, 15)])
please input a sentence (enter [exit] to exit.)
```

参考

- NER相关的论文整理在papers下
- https://github.com/scofield7419/sequence-labeling-BiLSTM-CRF
- 维特比解码器
- 最通俗易懂的BiLSTM-CRF模型中的CRF层介绍
- CRF Layer on the Top of BiLSTM - 1