DeepKE 实验环境搭建及操作教程

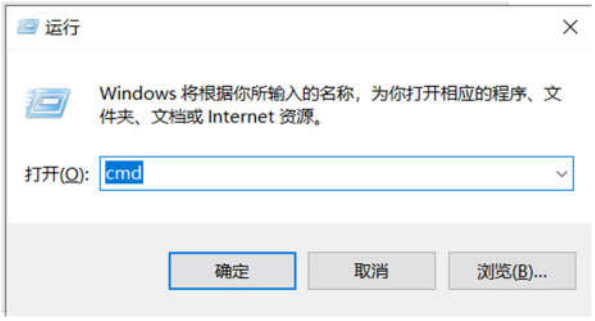
DeepKE 是一个支持低资源、长篇章、多模态的知识抽取工具，学术界和工业界的用户可以定制输入的数据集和模型实现命名实体识别、关系抽取和属性抽取功能。

一、 准备工作：Anaconda 的安装与配置

下载地址：https://www.anaconda.com/download

参考教程：https://zhuanlan.zhihu.com/p/32925500

Anaconda 安装验证：
用键盘输入 win+R 键，并在任务框内输入 cmd 点击确定进入命令行界面。



在命令行界面输入 conda --version 命令，出现版本信息则证明安装成功。

```
C:\Users\lenovo>conda --version
conda 4.10.3
```

二、 使用 Anaconda 进行实验环境的配置

输入 conda create -n deepke python=3.8 创建虚拟环境。

```
E:\deepke>conda create -n deepke python=3.8
```

输入 conda activate deepke 激活虚拟环境，此时即可在虚拟环境内配置需要的包，而不对全局环境造成影响。

```
E:\deepke>conda activate deepke
```

输入 pip install deepke 安装 deepke 库，即可自动完成 deepke 及其所需依赖的安装。

```
(deepke) E:\deepke>pip install deepke
```

注：这样安装的 torch 库是 cpu 版本的，有 N 系显卡的同学如果有兴趣的话可以先安装 GPU 版本的 torch，之后再安装 deepke 库。使用 GPU 进行计算可以极大程度提高神经网络的运算速度。

参考链接：https://zhuanlan.zhihu.com/p/479848495

三、 关系抽取实验

本次实验主要是利用 deepke 完成关系抽取任务，包括模型训练和实例测试。
关系抽取任务即在给定两个实体，自动抽取两者之间的关系，比如电影《长腿叔叔》和孔正锡之间的关系是“导演”。

目录 deepke\DeepKE-main\example\re\standard\data\origin 下存放本次实验使用的数据集，大家可以将其中的文件换成自己的数据训练特定的关系抽取模型。
关系文件：

head_type	tail_type	relation	index
None	None	None	0
影视作品	人物	导演	1
人物	国家	国籍	2
人物	地点	祖籍	3
电视综艺	人物	主持人	4
人物	地点	出生地	5
景点	城市	所在城市	6
歌曲	音乐专辑	所属专辑	7
网络小说	网站	连载网站	8
影视作品	企业	出品公司	9
人物	学校	毕业院校	10

训练集文件：

sentence	relation	head	head_offse	tail	tail_offset
明朝末年	出生地	黄得功	8	安徽合肥	16
《逐风行》	连载网站	逐风行	1	纵横中文	12
谢万松，5	出生地	谢万松	0	湖北省武	8
《娘家的	导演	娘家的故	1	张玲	11
九玄珠是	连载网站	九玄珠	0	纵横中文	5
《下半生	连载网站	下半生留	1	17k小说网	26
个人简介	国籍	梁信强	4	中国	20
李春英，7	国籍	李春英	0	中国	49
《穿越之	连载网站	穿越之鸣	1	17k小说网	15
王润泽1，	毕业院校	王润泽	0	四川外语	31
《舌尖上	导演	舌尖上的	1	陈晓卿	76
《死亡塔》	导演	飞龙猛将	17	元奎	37
《如果我	出品公司	如果我爱	1	海润影视	9
个人信息	国籍	伊万-佩恩	7	法国	26
《是我不	所属专辑	是我不小	1	相信你总	24
覃林盛，3	毕业院校	覃林盛	0	华中理工	78
2015年10	所在城市	龙泉寺	14	北京	12
0后的第三	主持人	电玩快打	28	纳豆	10
"观察人十	国籍	金正恩	54	朝鲜	13

1. 模型训练
首先需要修改配置文件，进入目录 deepke\DeepKE-main\example\re\standard\conf，
打开 train.yaml

```yaml
seed: 1

use_gpu: False
gpu_id: 0

epoch: 50
batch_size: 32
learning_rate: 3e-4
lr_factor: 0.7 # 学习率的衰减率
lr_patience: 3 # 学习率衰减的等待epoch
weight_decay: 1e-3 # L2正则

early_stopping_patience: 6

train_log: True
log_interval: 10
show_plot: False
only_comparison_plot: False
plot_utils: matplot  # [matplot, tensorboard]

predict_plot: False

use_multi_gpu: False
gpu_ids: 0,1
```

use_gpu:是否使用 GPU，设置为 False。
epoch：使用所有训练数据进行训练的训练轮数，可以适当调小
batch_size:一批次训练的数据量大小，可以根据电脑内存情况适当调小。
其他参数同样会对训练过程和结果造成影响，大家可以查阅资料进行了解并尝试修改。

打开 config.yaml

```yaml
cwd: ???


defaults:
  - hydra/output: custom
  - preprocess
  - train
  - embedding
  - predict
  - model: rnn   # [cnn, rnn, transformer, capsule, gcn, lm]
```

需要修改的参数为-model，默认为 lm，即 bert 等语言模型，我们将参数修改为硬件要求较低的 rnn，有兴趣的同学可以尝试其它模型，比较不同模型的性能差距。

进入命令行界面后，输入盘符（如 E:)进入项目所在硬盘后，输入项目所在的绝对路径（如 E:\deepke\DeepKE-main\example\re\standard）进入项目所在的目录。

```
C:\Users\lenovo>E:

E:\>cd E:\deepke\DeepKE-main\example\re\standard

E:\deepke\DeepKE-main\example\re\standard>
```

激活虚拟环境。

```
E:\deepke\DeepKE-main\example\re\standard>conda activate deepke

(deepke) E:\deepke\DeepKE-main\example\re\standard>
```

输入 python run.py 训练关系抽取模型。

```
(deepke) E:\deepke\DeepKE-main\example\re\standard>python run.py
```

输入 3，不可视化结果。

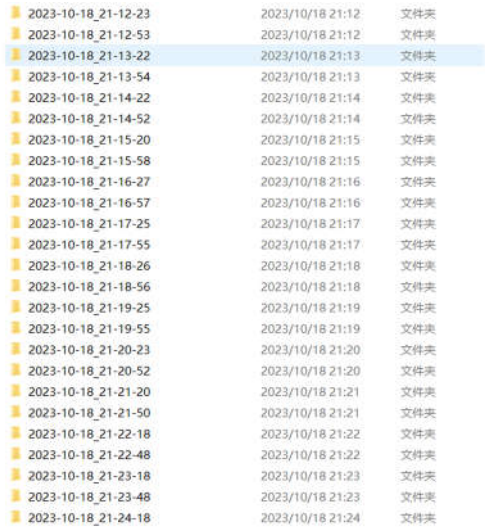
```
log_interval: 10
show_plot: false
only_comparison_plot: false
plot_utils: matplot
predict_plot: false
use_multi_gpu: false
gpu_ids: 0,1
vocab_size: ???
word_dim: 60
pos_size: 62
pos_dim: 10
dim_strategy: sum
num_relations: 11
fp: xxx/checkpoints/2019-12-03_17-35-30/cnn_epoch21.pth
model_name: lm
lm_file: bert-base-chinese
num_hidden_layers: 1
type_rnn: LSTM
input_size: 768
hidden_size: 100
num_layers: 1
dropout: 0.3
bidirectional: true
last_layer_hn: true

wandb: (1) Create a W&B account
wandb: (2) Use an existing W&B account
wandb: (3) Don't visualize my results
←[34m←[1mwandb←[0m: Enter your choice:
```

接下来等候模型训练完成即可，并且我们可以从命令行输出的日志信息得到模型在各轮训练的准确率等性能指标。

```
[2023-10-17 19:33:29,169][deepke.relation_extraction.standard.tools.trainer][INFO] - Train Epoch 50: Acc: 99.72%
macro metrics: [p: 0.9962, r:0.9937, f1:0.9949]
```

训练完成后，我们可以在 checkpoints 子目录下找到各轮训练好的参数文件



2. 实例测试

打开配置目录下的 predict.yaml 文件，修改下面的模型路径为想要测试的模型的绝对路径。

```
# the path of the model / checkpoint to be used
fp: 'E:\deepke\DeepKE-main\example\re\standard\checkpoints\2023-10-18_21-24-18\rnn_epoch25.pth'
```

输入 python predict.py 开始实例测试。

```
(deepke) E:\deepke\DeepKE-main\example\re\standard>python predict.py
```

实力测试包含两种，一种是使用给定的样例，另一种是自己输入句子和实体进行关系抽取。

输入 y 即可使用范例进行测试。

```
是否使用范例[y/n]，退出请输入：exit .... y
```

得到正确的抽取结果。

```
sentence: 歌曲《人生长路》出自刘德华国语专辑《男人的爱》，由李泉作词作曲，2001年出行发版
chinese_split: True
replace_entity_with_type:  False
replace_entity_with_scope: True
tokens:    ['歌曲', '《', 'TAIL', '》', '出自', '刘德华', '国语专辑', '《', 'HEAD', '》', ',', '由', '李泉', '作词', '
作曲', ',', '2001', '年', '出行', '发版']
token2idx: [216, 16, 8, 17, 1969, 462, 1, 16, 7, 17, 12, 194, 4158, 359, 362, 12, 1877, 46, 1, 1]
length:    20
head_idx:  8
tail_idx:  2
[2023-10-21 20:58:12,654][__main__][INFO] - device: cpu
[2023-10-21 20:58:12,812][__main__][INFO] - model name: rnn
[2023-10-21 20:58:12,813][__main__][INFO] -
 BiLSTM(
  (embedding): Embedding(
    (wordEmbed): Embedding(4687, 60, padding_idx=0)
    (entityPosEmbed): Embedding(62, 60, padding_idx=0)
    (attribute_keyPosEmbed): Embedding(62, 60, padding_idx=0)
    (layer_norm): LayerNorm((60,), eps=1e-05, elementwise_affine=True)
  )
  (bilstm): RNN(
    (rnn): LSTM(60, 75, num_layers=2, batch_first=True, dropout=0.3, bidirectional=True)
  )
  (fc): Linear(in_features=150, out_features=11, bias=True)
  (dropout): Dropout(p=0.3, inplace=False)
)
[2023-10-21 20:58:13,616][__main__][INFO] - "男人的爱" 和 "人生长路" 在句中关系为："所属专辑"，置信度为0.96。
```

输入 n 则可手动输入进行测试，我们可以使用数据集中的例子，也可以自己输入例子，但需要注意抽取的关系只会在给定的关系类别中。

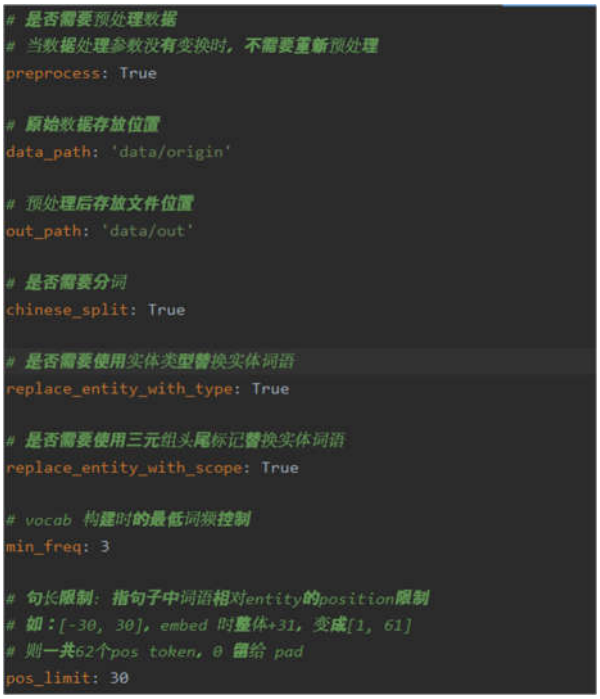
```
是否使用范例[y/n]，退出请输入：exit .... n
请输入句子：孔正锡，导演，2005年以一部温馨的爱情电影《长腿叔叔》敲开电影界大门
请输入句中需要预测关系的头实体：长腿叔叔
请输入头实体类型：影视作品
请输入句中需要预测关系的尾实体：孔正锡
请输入尾实体类型：人物
[2023-10-17 20:20:04,945][__main__][INFO] - "长腿叔叔" 和 "孔正锡" 在句中关系为："导演"，置信度为1.00。
```

当我们在手动输入测试样例时，如果不输入头实体类型和尾实体类型，那么关系抽取的结果会是错误的。

```
是否使用范例[y/n]，退出请输入：exit .... n
请输入句子：孔正锡，导演，2005年以一部温馨的爱情电影《长腿叔叔》敲开电影界大门
请输入句中需要预测关系的头实体：长腿叔叔
请输入头实体类型：
请输入句中需要预测关系的尾实体：孔正锡
请输入尾实体类型：
[2023-10-21 21:17:41,506][__main__][INFO] - "长腿叔叔" 和 "孔正锡" 在句中关系为："出品公司"，置信度为0.40。
```

这是因为我们在训练模型时将实体替换为实体类型进行训练，所以在测试时需要

输入实体类型的信息。可以修改配置目录下的 preprocess.yaml 预处理文件。将是否需要使用实体类型替换实体词语的配置改为 False，选择直接使用实体而不是实体类型进行实验。



修改配置后，依照上面的步骤重新进行模型训练，并将 predict.yam 的模型路径修改为新训练得到的模型的路径，重新进行实例的测试即可得到正确的结果。