DeepKE 实验环境搭建及操作教程

一、 准备工作：Anaconda 的安装与配置

```
C:\Users\34418>conda --version
conda 25.3.1
```

二、 使用 Anaconda 进行实验环境的配置

1、创建虚拟环境。

```
## Package Plan ##

  environment location: C:\Users\34418\.conda\envs\deepke

  added / updated specs:
    - python=3.8


The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    ca-certificates-2025.4.26  |       h4c7d964_0         149 KB  https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/ce
    openssl-3.5.0              |       ha4e3fda_1         8.6 MB  https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/ce
    ------------------------------------------------------------
                                           Total:         8.8 MB

done
#
# To activate this environment, use
#
#     $ conda activate deepke
#
# To deactivate an active environment, use
#
#     $ conda deactivate
```

2、激活虚拟环境

```
C:\Users\34418>conda activate deepke

(deepke) C:\Users\34418>
```

3、安装 deepke 库

```
Successfully installed GitPython-3.1.44 Jinja2-3.1.2 MarkupSafe-2.1.5 absl-py-2.2.2 aiohappyeyeballs-2.4.4 aiohttp-3.10.
11 aiosignal-1.3.1 antlr4-python3-runtime-4.8 asttokens-3.0.0 async-timeout-5.0.1 attrs-25.3.0 backcall-0.2.0 cachetools
-4.2.4 certifi-2025.4.26 charset-normalizer-3.4.2 click-8.1.8 colorama-0.4.6 configparser-7.1.0 cycler-0.12.1 datasets-2
.13.2 decorator-5.2.1 deepke-2.2.7 dill-0.3.6 docker-pycreds-0.4.0 executing-2.2.0 filelock-3.16.1 frozenlist-1.5.0 fssp
ec-2025.3.0 gitdb-4.0.12 google-auth-1.35.0 google-auth-oauthlib-0.4.6 grpcio-1.70.0 huggingface_hub-0.11.0 hydra-core-1
.0.6 idna-3.10 importlib-metadata-8.5.0 importlib-resources-6.4.5 ipdb-0.13.11 ipython-8.12.3 jedi-0.19.2 jieba-0.42.1 j
oblib-1.4.2 kiwisolver-1.4.7 markdown-3.7 matplotlib-3.4.1 matplotlib-inline-0.1.7 multidict-6.1.0 multiprocess-0.70.14
nltk-3.8 numpy-1.21.0 oauthlib-3.2.2 omegaconf-2.0.6 openai-0.28.0 opt-einsum-3.3.0 packaging-25.0 pandas-2.0.3 parso-0.
8.4 pathtools-0.1.2 pickleshare-0.7.5 pillow-10.4.0 promise-2.3 prompt-toolkit-3.0.51 propcache-0.2.0 protobuf-3.20.1 ps
util-7.0.0 pure-eval-0.2.3 pyarrow-17.0.0 pyasn1-0.6.1 pyasn1-modules-0.4.2 pygments-2.19.1 pyhocon-0.3.60 pyparsing-3.1
.4 python-dateutil-2.9.0.post0 pytorch-crf-0.7.2 pytz-2025.2 pyyaml-6.0.2 regex-2024.11.6 requests-2.32.3 requests-oauth
lib-2.0.0 rsa-4.9.1 scikit-learn-0.24.1 scipy-1.10.1 sentry-sdk-2.27.0 seqeval-1.2.2 shortuuid-1.0.13 six-1.17.0 smmap-5
.0.2 stack-data-0.6.3 subprocess32-3.5.4 tensorboard-2.4.1 tensorboard-plugin-wit-1.8.1 tensorboardX-2.5.1 termcolor-2.4
.0 threadpoolctl-3.5.0 tokenizers-0.13.3 tomli-2.2.1 torch-1.11.0 tqdm-4.66.1 traitlets-5.14.3 transformers-4.26.0 typin
g-extensions-4.13.2 tzdata-2025.2 ujson-5.6.0 urllib3-2.2.3 wandb-0.12.7 wcwidth-0.2.13 werkzeug-3.0.6 xxhash-3.5.0 yarl
-1.15.2 yaspin-2.5.0 zipp-3.20.2

[notice] A new release of pip is available: 24.0 -> 25.0.1
[notice] To update, run: python.exe -m pip install --upgrade pip
```

三、 关系抽取实验

1. 模型训练

参数修改

```
use_gpu: False
gpu_id: 0
```

```
  - model: rnn    # [cnn, rnn, transformer,
  capsule, gcn, lm]
```

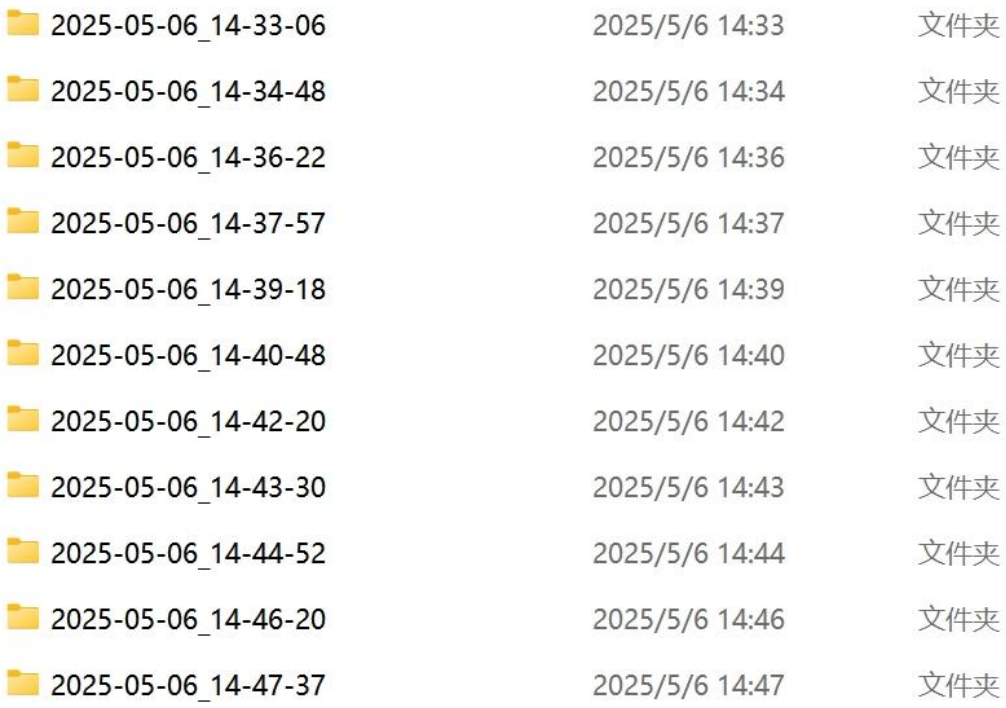
激活虚拟环境。

```
C:\Users\34418>D:

D:\>cd D:\KnoledgeEngineering\lab10_deepke\DeepKE-main\example\re\standard

D:\KnoledgeEngineering\lab10_deepke\DeepKE-main\example\re\standard>conda activate deepke
```

输入 python run.py 训练关系抽取模型。

```
[2025-05-06 15:51:53,098][__main__][INFO] - this model save path: D:\KnoledgeEngineering\lab10_deepke\DeepKE-main\exampl
e\re\standard\checkpoints\2025-05-06_15-02-49\rnn_epoch21.pth
[2025-05-06 15:51:53,098][__main__][INFO] - total 50 epochs, best(valid macro f1) epoch: 22, this epoch macro f1: 0.4643
[2025-05-06 15:51:53,098][__main__][INFO] - =====end of training====
[2025-05-06 15:51:53,098][__main__][INFO] -
[2025-05-06 15:51:53,099][__main__][INFO] - =====start test performance====
[2025-05-06 15:51:57,426][deepke.relation_extraction.standard.tools.trainer][INFO] - Test Data: [1000/1000](100%)
Loss: 0.210479
[2025-05-06 15:51:57,427][deepke.relation_extraction.standard.tools.trainer][INFO] - Test Data: Acc: 95.00%     macro me
trics: [p: 0.9554, r:0.9500, f1:0.9493]
[2025-05-06 15:51:57,429][__main__][INFO] - =====ending====

wandb: Waiting for W&B process to finish, PID 22496... (success).
wandb: Run history:
wandb:    test_loss
wandb:   train_loss
wandb:   valid_loss
wandb:
wandb: Run summary:
wandb:    test_loss 0.21048
wandb:   train_loss 0.00562
wandb:   valid_loss 0.43442
wandb:
wandb: You can sync this run to the cloud by running:
wandb: wandb sync D:\KnoledgeEngineering\lab10_deepke\DeepKE-main\example\re\standard\logs\2025-05-06_14-31-11\wandb\off
line-run-20250506_143128-85h17xa5
wandb: Find logs at: .\logs\2025-05-06_14-31-11\wandb\offline-run-20250506_143128-85h17xa5\logs\debug.log
wandb:
```

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

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:
'D:\KnoledgeEngineering\lab10_deepke\DeepKE-main\example\re\standard\checkpoints\2025-05-06_15-51-53\rnn_epoch50.pth'
```

使用给定的样例

```
是否使用范例[y/n]，退出请输入：exit .... y
[2025-05-06 21:47:46,366][__main__][INFO] - start sentence preprocess...
[2025-05-06 21:47:46,366][__main__][INFO] -
```

```
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
[2025-05-06 21:47:46,367][__main__][INFO] - device: cpu
[2025-05-06 21:47:46,376][__main__][INFO] - model name: rnn
[2025-05-06 21:47:46,377][__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)
)
[2025-05-06 21:47:46,405][__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0.97。
```

输入 n 则可手动输入进行测试

```
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
```

```
sentence: 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
chinese_split: True
replace_entity_with_type:  True
replace_entity_with_scope: True
tokens:    ['TAIL_人物', ', ', '导演', ', ', '2005', '年', '以', '一部', '温馨', '的', '爱情', '电影', '《', 'HEAD_影视
作品', '》', '敲开', '电影界', '大门']
token2idx: [52, 13, 144, 13, 379, 50, 284, 31, 3422, 30, 560, 488, 18, 51, 20, 1, 3439, 1]
length:    18
head_idx:  13
tail_idx:  0
[2025-05-06 16:31:51,452][__main__][INFO] - device: cpu
[2025-05-06 16:31:51,460][__main__][INFO] - model name: rnn
[2025-05-06 16:31:51,461][__main__][INFO] -
 BiLSTM(
  (embedding): Embedding(
    (wordEmbed): Embedding(4701, 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)
)
[2025-05-06 16:31:51,492][__main__][INFO] - "长腿叔叔" 和 "孔正锡" 在句中关系为："导演"，置信度为0.98。
```

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：
```

```
sentence: 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
chinese_split: True
replace_entity_with_type:  False
replace_entity_with_scope: True
tokens:    ['TAIL', ', ', '导演', ', ', '2005', '年', '以', '一部', '温馨', '的', '爱情', '电影', '《', 'HEAD', '》', '
敲开', '电影界', '大门']
token2idx: [8, 13, 144, 13, 379, 50, 284, 31, 3422, 30, 560, 488, 18, 7, 20, 1, 3439, 1]
length:    18
head_idx:  13
tail_idx:  0
[2025-05-06 16:34:14,624][__main__][INFO] - device: cpu
[2025-05-06 16:34:14,642][__main__][INFO] - model name: rnn
[2025-05-06 16:34:14,643][__main__][INFO] -
 BiLSTM(
  (embedding): Embedding(
    (wordEmbed): Embedding(4701, 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)
)
[2025-05-06 16:34:14,677][__main__][INFO] - "长腿叔叔" 和 "孔正锡" 在句中关系为："主持人"，置信度为0.50。
```

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。

```
# 是否需要使用实体类型替换实体词语
replace_entity_with_type: False
```

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。

```
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：
```

```
[2025-05-06 21:46:38,004][__main__][INFO] -
sentence: 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
chinese_split: True
replace_entity_with_type:  False
replace_entity_with_scope: True
tokens:    ['TAIL', ', ', '导演', ', ', '2005', '年', '以', '一部', '温馨', '的', '爱情', '电影', '《', 'HEAD', '》', '
敲开', '电影界', '大门']
token2idx: [8, 12, 136, 12, 365, 46, 270, 27, 3408, 26, 546, 474, 16, 7, 17, 1, 3425, 1]
length:    18
head_idx:  13
tail_idx:  0
[2025-05-06 21:46:38,007][__main__][INFO] - device: cpu
[2025-05-06 21:46:38,016][__main__][INFO] - model name: rnn
[2025-05-06 21:46:38,017][__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)
)
[2025-05-06 21:46:38,052][__main__][INFO] - "长腿叔叔" 和 "孔正锡" 在句中关系为："导演"，置信度为0.99。
```