什么是conda？

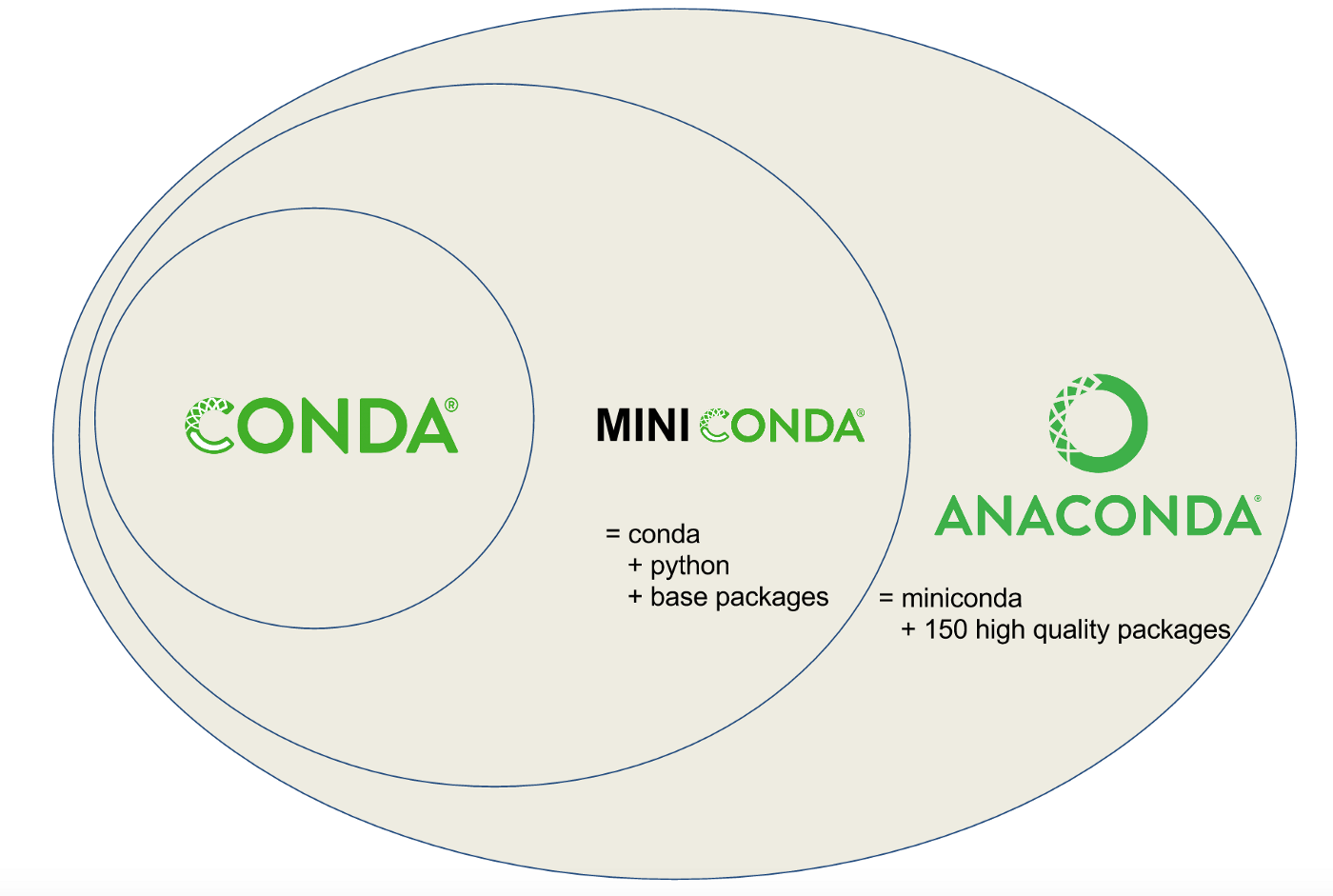
Conda是一个开源跨平台语言无关的包管理与环境管理系统。[[1]](#footnote-1)

Conda可以快速的安装、更新、运行软件包及其相关的依赖项。[[2]](#footnote-2)

Conda可以在你本地计算机上创建、加载、保存、切换不同项目特定的软件环境。

尽管Conda是为Python设计，但是其可以为多种编程语言打包并分发软件，例如R、Ruby等。

简而言之，Conda作为「包管理器」，其功能就是帮助你找到并且安装各种包。而作为「环境管理器」，其功能是在你需要不同版本的Python来支持你的项目或者用到的包时，Conda可以帮助我们管理Python的版本，而不至于出现冲突。

Conda，Miniconda，Anaconda

下图展示了三者之间的关系。[[3]](#footnote-3)

通常我们安装Miniconda就足够了，其一并包含了conda、python和一些基础的软件包。安装整个Anaconda太大了，并且Miniconda就可以为不同项目创建不同的环境。

Conda有什么用？

可以方便我们安装一些软件，比如Tensorflow[[4]](#footnote-4)。

Conda跨平台，在不同操作系统、硬件环境都可以用。

当不能用conda指令直接安装时，可以在Conda环境里直接用其他包管理工具，比如pip。

怎么安装？

安装conda，Miniconda[[5]](#footnote-5)，Anaconda[[6]](#footnote-6)的教程已经很多了，这里不再赘述，需要请自行搜索。

之后记得将conda添加到环境变量[[7]](#footnote-7)$PATH[[8]](#footnote-8)，这样才可以在命令行中使用conda命令。

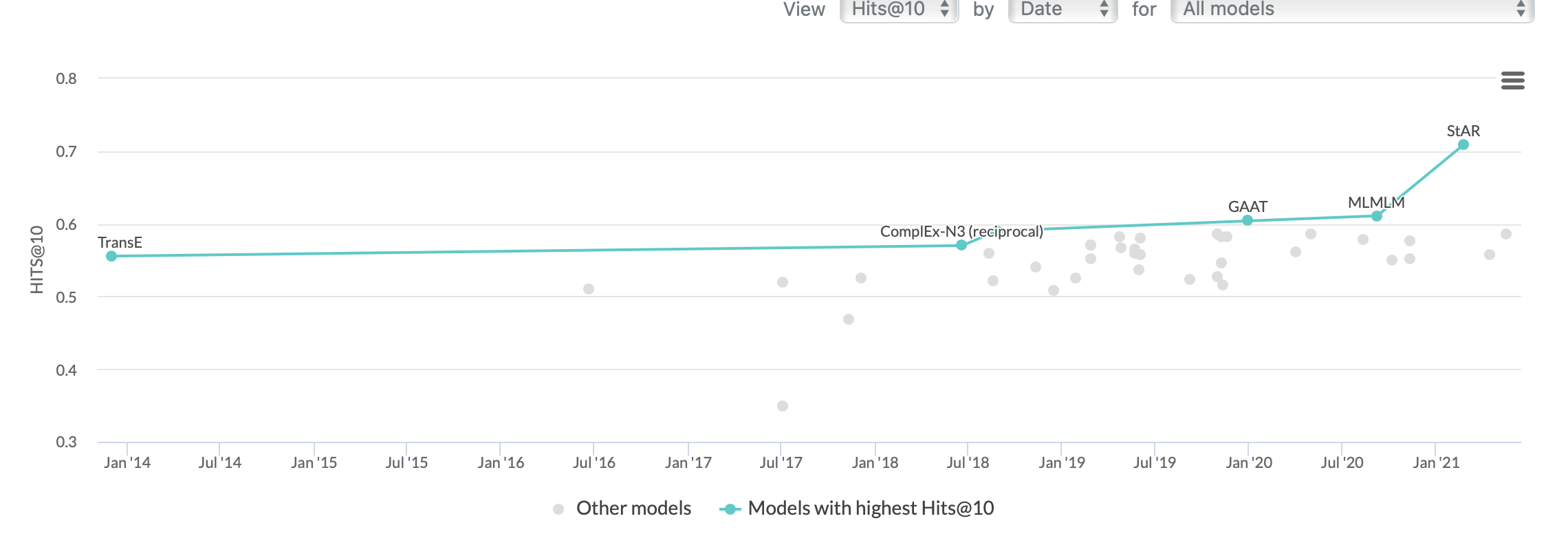
如果之前安装过python，可能会出现一些麻烦的问题[[9]](#footnote-9)。需要针对性查找解决方案。

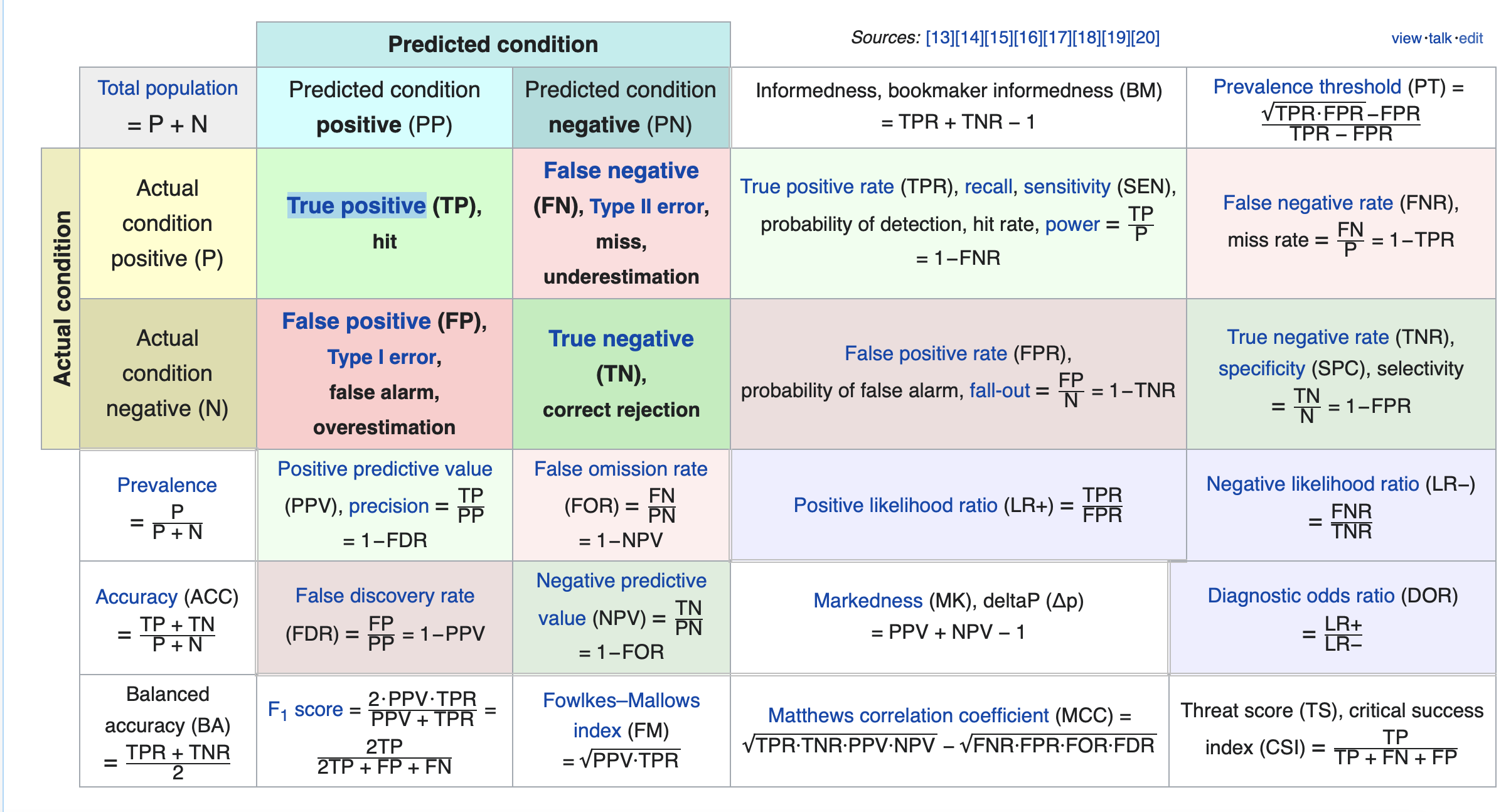
Conda命令

常见命令[[10]](#footnote-10)介绍了conda指令的入门用法，包括conda管理、环境管理、python管理等。

我们先尝试复现几个链接预测模型[[11]](#footnote-11)。

StAR模型[[12]](#footnote-12)[[13]](#footnote-13)

吉林大学王博同学[[14]](#footnote-14)在2021年1月的工作。在2021-6-23仍然SOTA，击中率[[15]](#footnote-15)很高。



第一步：安装anaconda[[16]](#footnote-16)，调整环境变量[[17]](#footnote-17)，使终端中 python 指向安装的anaconda目录下的python[[18]](#footnote-18)，并看看pip对应python是否正确。

第二步：按照github仓库readme[[19]](#footnote-19)，安装环境。

2. Installing requirement packages

conda create -n StAR python=3.6

source activate StAR

pip install numpy torch tensorboardX tqdm boto3 requests regex sacremoses sentencepiece

2.1 Optional package (for mixed float Computation)

git clone https://github.com/NVIDIA/apex

cd apex

pip install -v --no-cache-dir --global-option="--cpp\_ext" --global-option="--cuda\_ext" ./

第三步：调整数据集

数据集：WN18RR[[20]](#footnote-20), FB15k-237[[21]](#footnote-21), UMLS[[22]](#footnote-22)

训练和测试都在./data目录下；在原始开发集上验证的代价大，我们在训练时在开发集的子集上进行验证；WN18RR开发集的子集在./data/WN18RR下，叫new\_dev.dict. 也可以用下面的指令可以生成这个测试集的子集。（文件已经包含）

论文把另一个数据集 NELL-One[[23]](#footnote-23) 格式调整为上面三个数据集的形式。

CUDA\_VISIBLE\_DEVICES=0 \

python get\_new\_dev\_dict.py \

--model\_class bert \

--weight\_decay 0.01 \

--learning\_rate 5e-5 \

--adam\_epsilon 1e-6 \

--max\_grad\_norm 0. \

--warmup\_proportion 0.05 \

--do\_train \

--num\_train\_epochs 7 \

--dataset WN18RR \

--max\_seq\_length 128 \

--gradient\_accumulation\_steps 4 \

--train\_batch\_size 16 \

--eval\_batch\_size 128 \

--logging\_steps 100 \

--eval\_steps -1 \

--save\_steps 2000 \

--model\_name\_or\_path bert-base-uncased \

--do\_lower\_case \

--output\_dir ./result/WN18RR\_get\_dev \

--num\_worker 12 \

--seed 42 \

运行下面的指令得到调整格式的数据。（文件已经包含）

python reformat\_nell\_one.py --data\_dir path\_to\_downloaded --output\_dir ./data/NELL\_standard

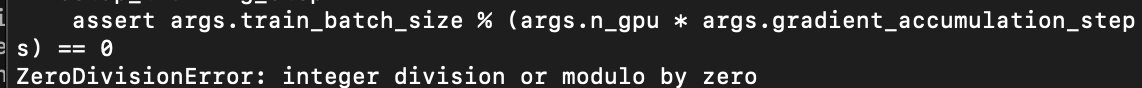
第四步：训练并测试模型

运行下面的代码就可以复现论文结果。注意，所有的 eval\_steps 设为 -1 ，在训练时不验证并保存checkpoint，因为标准的开发集太费时。

4.1 WN18RR

（缺少的模块手动补全、下载数据慢）图像

（没有GPU运行不了）



4.2 FB15k-237

CUDA\_VISIBLE\_DEVICES=0 \

python run\_link\_prediction.py \

--model\_class roberta \

--weight\_decay 0.01 \

--learning\_rate 1e-5 \

--adam\_betas 0.9,0.98 \

--adam\_epsilon 1e-6 \

--max\_grad\_norm 0. \

--warmup\_proportion 0.05 \

--do\_train --do\_eval \

--do\_prediction \

--num\_train\_epochs 7 \

--dataset WN18RR \

--max\_seq\_length 128 \

--gradient\_accumulation\_steps 4 \

--train\_batch\_size 16 \

--eval\_batch\_size 128 \

--logging\_steps 100 \

--eval\_steps 4000 \

--save\_steps 2000 \

--model\_name\_or\_path roberta-large \

--output\_dir ./result/WN18RR\_roberta-large \

--num\_worker 12 \

--seed 42 \

--cls\_method cls \

--distance\_metric euclidean \

CUDA\_VISIBLE\_DEVICES=0 \

python run\_link\_prediction.py \

--model\_class roberta \

--weight\_decay 0.01 \

--learning\_rate 1e-5 \

--adam\_betas 0.9,0.98 \

--adam\_epsilon 1e-6 \

--max\_grad\_norm 0. \

--warmup\_proportion 0.05 \

--do\_train --do\_eval \

--do\_prediction \

--num\_train\_epochs 7. \

--dataset FB15k-237 \

--max\_seq\_length 100 \

--gradient\_accumulation\_steps 4 \

--train\_batch\_size 16 \

--eval\_batch\_size 128 \

--logging\_steps 100 \

--eval\_steps -1 \

--save\_steps 2000 \

--model\_name\_or\_path roberta-large \

--output\_dir ./result/FB15k-237\_roberta-largel \

--num\_worker 12 \

--seed 42 \

--fp16 \

--cls\_method cls \

--distance\_metric euclidean \

4.3 UMLS

CUDA\_VISIBLE\_DEVICES=0 \

python run\_link\_prediction.py \

--model\_class roberta \

--weight\_decay 0.01 \

--learning\_rate 1e-5 \

--adam\_betas 0.9,0.98 \

--adam\_epsilon 1e-6 \

--max\_grad\_norm 0. \

--warmup\_proportion 0.05 \

--do\_train --do\_eval \

--do\_prediction \

--num\_train\_epochs 20 \

--dataset UMLS \

--max\_seq\_length 16 \

--gradient\_accumulation\_steps 1 \

--train\_batch\_size 16 \

--eval\_batch\_size 128 \

--logging\_steps 100 \

--eval\_steps -1 \

--save\_steps 200 \

--model\_name\_or\_path roberta-large \

--output\_dir ./result/UMLS\_model \

--num\_worker 12 \

--seed 42 \

--cls\_method cls \

--distance\_metric euclidean

4.4 NELL-One

CUDA\_VISIBLE\_DEVICES=0 \

python run\_link\_prediction.py \

--model\_class bert \

--do\_train --do\_eval \usepacka--do\_prediction \

--warmup\_proportion 0.1 \

--learning\_rate 5e-5 \

--num\_train\_epochs 8. \

--dataset NELL\_standard \

--max\_seq\_length 32 \

--gradient\_accumulation\_steps 1 \

--train\_batch\_size 16 \

--eval\_batch\_size 128 \

--logging\_steps 100 \

--eval\_steps -1 \

--save\_steps 2000 \

--model\_name\_or\_path bert-base-uncased \

--do\_lower\_case \

--output\_dir ./result/NELL\_model \

--num\_worker 12 \

--seed 42 \

--fp16 \

--cls\_method cls \

--distance\_metric euclidean

5. StAR\_Self-Adp

5.1 数据处理

先得到RotatE模型[[24]](#footnote-24)。（什么叫得到？会不会是这一步没做引起的报错）

顺序运行下面的指令，得到StAR\_Self-Adp的训练数据。

运行./StaR目录下的run\_get\_unsembele\_data.py

CUDA\_VISIBLE\_DEVICES=0 python run\_get\_unsemble\_data.py \

--dataset WN18RR \

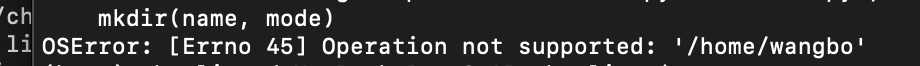
--model\_class roberta \

--model\_name\_or\_path /home/wangbo/workspace/StAR/result/WN18RR\_roberta-large \

--output\_dir /home/wangbo/workspace/StAR/result/WN18RR\_roberta-large \

--seed 42 \

--fp16

注意这里要修改代码中的路径名，否则用的是作者的路径名

运行./RoratE目录下的run.py

CUDA\_VISIBLE\_DEVICES=0 python run.py \

--cuda --init models/RotatE\_wn18rr\_0 \

--test\_batch\_size 16 \

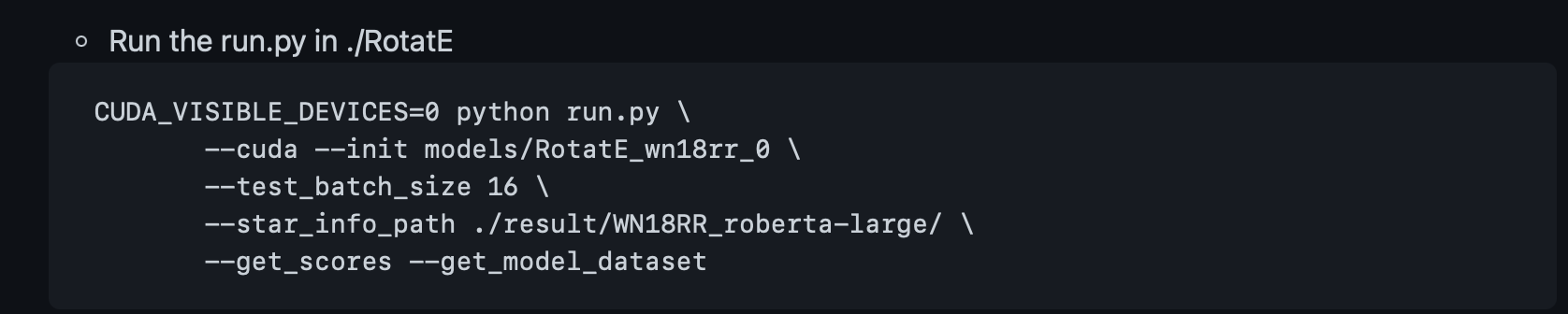
--star\_info\_path ./result/WN18RR\_roberta-large/ \

--get\_scores --get\_model\_dataset

错误

1.

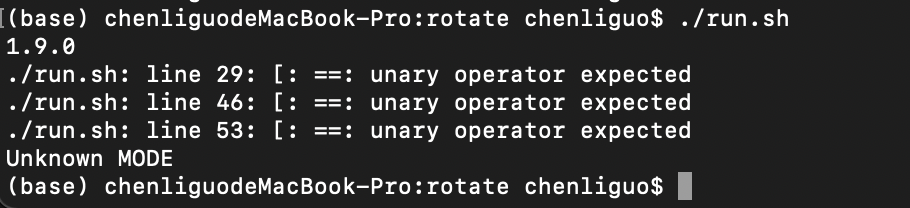
没有./RotatE目录，而是./rotate



目录下也没有run.py文件

有./rotate/run.sh

使用mac OS终端直接运行报错



也有./rotate/codes/run.py

运行发生错误2.

修改后还是发生错误图像

2.

./rotate/codes/model.py 893行的三个import应该删除

from \_\_future\_\_ import absolute\_import

from \_\_future\_\_ import division

from \_\_future\_\_ import print\_function

否则报错

SyntaxError: from \_\_future\_\_ imports must occur at the beginning of the file

1. https://zh.wikipedia.org/wiki/Conda [↑](#footnote-ref-1)
2. <https://towardsdatascience.com/managing-project-specific-environments-with-conda-b8b50aa8be0e> [↑](#footnote-ref-2)
3. https://planemo.readthedocs.io/en/latest/index.html [↑](#footnote-ref-3)
4. https://www.tensorflow.org [↑](#footnote-ref-4)
5. https://docs.conda.io/en/latest/miniconda.html [↑](#footnote-ref-5)
6. https://docs.anaconda.com/anaconda/install/ [↑](#footnote-ref-6)
7. https://zh.wikipedia.org/wiki/环境变量 [↑](#footnote-ref-7)
8. https://en.wikipedia.org/wiki/PATH\_(variable) [↑](#footnote-ref-8)
9. https://zhuanlan.zhihu.com/p/61717000 [↑](#footnote-ref-9)
10. https://www.cnblogs.com/sddai/p/10322603.html [↑](#footnote-ref-10)
11. https://paperswithcode.com/task/link-prediction [↑](#footnote-ref-11)
12. https://arxiv.org/pdf/2004.14781v2.pdf [↑](#footnote-ref-12)
13. https://github.com/wangbo9719/StAR\_KGC [↑](#footnote-ref-13)
14. bowang19@mails.jlu.edu.cn [↑](#footnote-ref-14)
15. https://en.wikipedia.org/wiki/Confusion\_matrix [↑](#footnote-ref-15)
16. https://www.anaconda.com/products/individual [↑](#footnote-ref-16)
17. https://www.zhihu.com/question/30941329 [↑](#footnote-ref-17)
18. https://jiyiren.github.io/2017/10/31/python\_environmet/ [↑](#footnote-ref-18)
19. https://github.com/wangbo9719/StAR\_KGC [↑](#footnote-ref-19)
20. <https://everest.hds.utc.fr/doku.php?id=en:transe>

    <https://paperswithcode.com/dataset/wn18rr>

    <https://zhuanlan.zhihu.com/p/133649429>

    <https://zh.wikipedia.org/wiki/WordNet> [↑](#footnote-ref-20)
21. <https://deepai.org/dataset/fb15k-237>

    https://blog.csdn.net/qq\_21097885/article/details/105620273 [↑](#footnote-ref-21)
22. <https://www.nlm.nih.gov/research/umls/index.html>

    <https://github.com/ningshixian/UMLs> [↑](#footnote-ref-22)
23. https://github.com/xwhan/One-shot-Relational-Learning

    <https://azft.alibaba.com/newspage/?id=69>

    <https://zhuanlan.zhihu.com/p/59646318> [↑](#footnote-ref-23)
24. https://github.com/DeepGraphLearning/KnowledgeGraphEmbedding [↑](#footnote-ref-24)