docker 指令

docker run --gpus all -p 主机端口:映射端口 -itd --name 容器名字 映像名字 /bin/bash 运行容器

docker ps -a 查看所有的容器

docker start ID/Name 启动一个已停止的容器

docker stop ID/Name 停止容器

docker restart ID/Name 重启容器

docker exec-it ID/Name /bin/bash 执行容器

docker rm ID/Name 删除容器

docker cp 本地文件路径 ID/Name:docker路径 拷贝文件（反过来适用）

docker ps -s 查看容器大小

docker ps -a 查看进程

docker rmi ImageID/Name 删除镜像

docker commit <container-id> <image-name> 把一个正在运行的容器变成一个新的镜像

docker save -o image\_name.tar <image-name> 该命令将镜像保存为tar文件

docker load -i tar文件 加载镜像文件到系统中

docker logs -f -t --tail=100 容器名字 查看容器后100行日志

docker commit -a "zujian-nlp" -m "creat a new docker image by zujian-nlp" 165bf5a33419 testnlp:v5 打包zujian-nlp容器为一个镜像名字：版本号为testnlp:v5

docker save -o testnlpV5.tar testnlp:v5 保存testnlp:v5镜像打包成压缩包

docker load -i testnlpV5.tar 加载镜像

linux 指令

du -h 查看目录及子目录大小

du -h --max-depth=1 | sort -rh 查看当前目录下各个文件夹大小

ll -h 查看当前目录文件大小

Kill -9 PID 杀死一个进程

nohup 后台不会断进程

ps aux | grep run.py 显示所有进程|获取py文件进程

watch -d -n 1 nvidia-smi 实时查看GPU

scp redis.conf root@k8s-node02:/home 传送文件

scp root@192.168.10.62:/zjh/nlp/nlptest.tar C:\Users\Administrator\Desktop

ll /usr/bin/ | grep python 查看原有软连接

rm -rf /usr/bin/python 删除原有软连接

ln -s /usr/bin/python3.7 /usr/bin/python 建立新软连接

lsof -i:100010 查看端口进程

ps aux | grep PID 查看进程ID程序

ping data.cs.purdue.edu 测试网络连接状况

Crontab 指令 定时操作

yum install crontabs 安装crontabs

crontab -l 查看定时任务

crontab -e 设置定时任务

\* 3 \* \* \* 执行的语句 # 每天3点执行

30 18 \* \* \* 执行的语句 # 每天18点30分执行

15 12 1 \* \* 执行的语句 # 每个月的1号的12点15分执行

45 11 1 3 \* 执行的语句 # 每年的3月1号的11点45分执行

30 16 \* \* 3 执行的语句 # 每周三的16点30分执行

30 14 \* \* \* python helloworld.py >> log.txt 14点30分执行py文件

Systemctl status crond.service 查看crontab状态

Systemctl start crond.service 启动crontab

systemctl stop crond.service 关闭crontab

Systemctl restart crond.service 重启conrtab

os.mkdir(save\_path) 创建单层目录

os.makedirs(save\_path, exist\_ok=True) 创建多级目录

时间

date +%D 07/27/23

date +%F 2023-07-27

date +%x 07/27/2023

date +%Y%m%d 20230727

防火墙

systemctl status firewalld 查看防火墙

systemctl stop firewalld 关闭防火墙

systemctl start firewalld 开启防火墙

firewall-cmd --list-port 查看防火墙端口

firewall-cmd --add-port=4300/tcp --permanent --zone=public 防火墙添加端口

firewall-cmd --reload 重启防火墙

python

**不同文件夹下调用文件和方法 app/run.py 调用 app/Pytorch\_Bert\_TextCNN\_CLS/predict.py**

this\_dir = os.path.abspath("Pytorch\_Bert\_TextCNN\_CLS")

sys.path.append(this\_dir)

sys.path.insert(0, this\_dir + '/Pytorch\_Bert\_TextCNN\_CLS/')

import predict

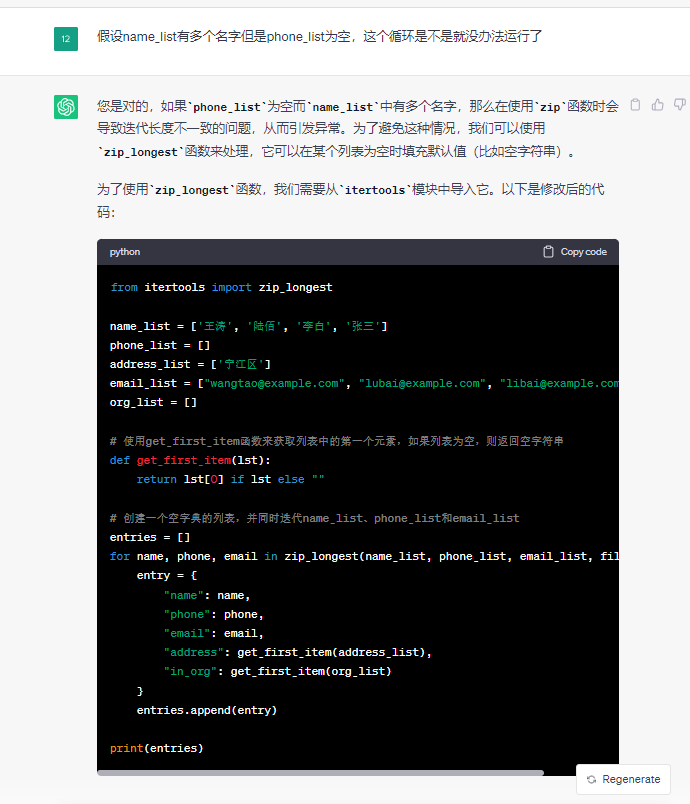
a = predict.classfier\_predict(content) 调用方法

**获取当前文件的路径**

script\_path = os.path.abspath(\_\_file\_\_)

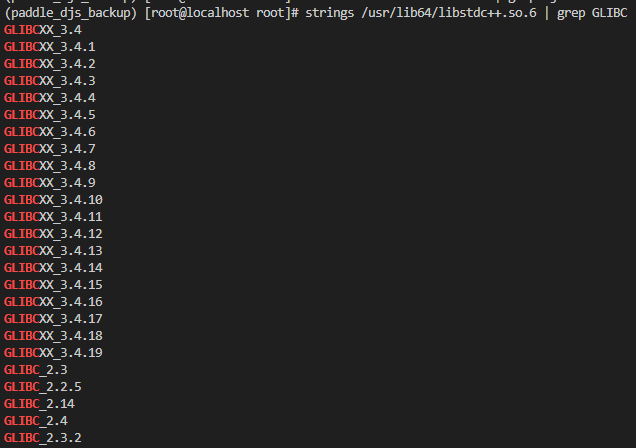
this\_dir = os.path.dirname(script\_path)

zip函数 zip\_longest



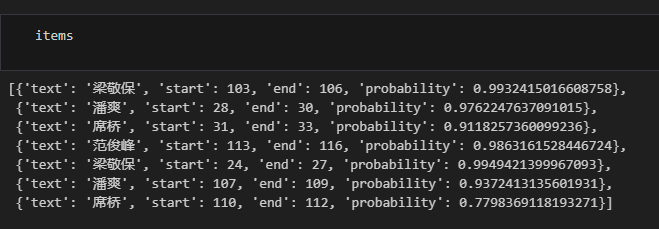
如果遇到/lib64/libstdc++.so.6: version `GLIBCXX\_3.4.21' not found

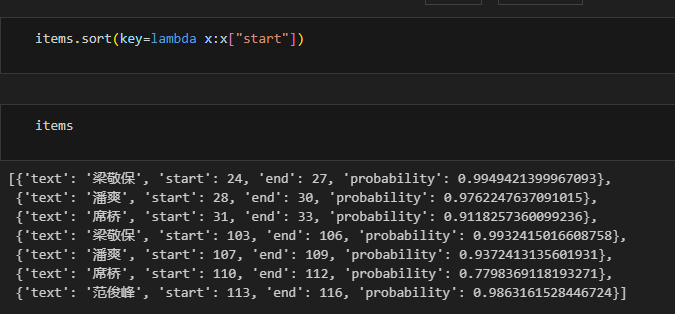
1. strings /usr/lib64/libstdc++.so.6 | grep GLIBC



1. find / -name "libstdc++.so\*"
2. cp /home/albay3/miniconda3/envs/paddle\_djs\_backup/lib/libstdc++.so.6.0.31 /usr/lib64/
3. ln -sf /lib64/libstdc++.so.6.0.31 /lib64/libstdc++.so.6

按照字典里的key排序





pip 指令

pip install package -i https://pypi.tuna.tsinghua.edu.cn/simple

清华：https://pypi.tuna.tsinghua.edu.cn/simple

阿里云：http://mirrors.aliyun.com/pypi/simple/

中国科技大学 https://pypi.mirrors.ustc.edu.cn/simple/

华中理工大学：http://pypi.hustunique.com/

山东理工大学：http://pypi.sdutlinux.org/

豆瓣：<http://pypi.douban.com/simple/>

pip install -h 帮助

pip install package --no-dependencies 忽略包的依赖性 ignore package dependencies

pip install package --no-deps

Conda 指令

conda create --name 新环境 --clone 旧环境 复制虚拟环境

conda remove --name 环境名 --all 删除环境

conda create --name 环境名 python=3.8 创建环境 指定版本

SQL 指令

truncate table 表名 清除table

Select \* from 表名 where id=27 从table选取所有id为27的数据

Update 表名 set “birthday”=26 where id=27 更改id为27数据列为生日的改成26

Git 指令

git init 初始化仓库

git add . 上传所有的文件

git commit -m “提交备注名” 上传备注

git remote add origin 加上仓库地址 连接远程仓库

git push origin master 提交本地文件到github项目中

Push和pull的时候可以加上 --allow-unrelated-histories

git pull origin master 同步github项目到本地

git status 查看git日志

git branch -m main master 把main重命名为master

git push origin --delete main 删除原来main分支

git reflog 查看所有日志

git reset --hard HEAD@{} 回退到当前版本

git branch -a 查看所有分支

git checkout -b project origin/master/origin 拉取远程到本地

git branch tmp id 创建一个临时分支合并上面最后一次提交的id

git checkout code 切换本地分支

git merge tmp 合并临时分支

git branch -D tmp 删除临时分支

