****

Robot4.00

安装说明文档

版本号 V1.8



**广州云趣信息科技有限公司**

**2025年8月**

**版本修订批准记录**

【历次版本修订记录，版本修订描述重要说明】

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **提交日期** | **版本号** | **版本修订描述** | **作者** | **批准人** |
| 2024-05-07 | 1.00 | 初始版本 | 雷思思 | 罗睦军 |
| 2024-05-11 | 1.1 | 增加6.2章节  Robot3.0到robot4.0升级步骤 | 雷思思 | 罗睦军 |
| 2024-06-06 | 1.2 | 增加6.3章节  使用普通用户启动robot | 雷思思 | 罗睦军 |
| 2024-07-10 | 1.3 | 增加faq6.3：数据库为mysql5.7版本时robot的驱动包替换 | 雷思思 | 罗睦军 |
| 2024-08-29 | 1.4 | 增加5.2章节第四步  查看robot许可证到期时间 | 雷思思 | 罗睦军 |
| 2025-07-10 | 1.5 | 增加“5.3实体提取接口部署”、  “5.4情绪识别接口部署”、“6.1若服务器性能不够，可以把python程序和java程序放在不同的服务器”章节 | 李楚茵 |  |
| 2025-07-17 | 1.6 | 修改了文本匹配、实体提取、情绪识别的默认端口 | 李楚茵 |  |
| 2025-07-23 | 1.7 | 增加“6.2文本匹配缓存写入文件时间、缓存最大句子数修改”章节 | 李楚茵 |  |
| 2025-08-04 | 1.8 | 修改“5.1依赖包安装”章节中“配置miniconda3环境变量”步骤，将环境变量写到~/.bashrc文件 | 李楚茵 |  |

**注：版本修订批准记录，如不需要可删除此页**

目录

[1 声明 5](#_Toc2454)

[1.1 关于版权 5](#_Toc25189)

[1.2 阅读约定 6](#_Toc5285)

[2 编写目的 7](#_Toc18697)

[3 部署前准备 7](#_Toc1762)

[4 安装包简介 7](#_Toc18173)

[5 安装步骤 8](#_Toc14758)

[5.1 依赖包安装 8](#_Toc10476)

[第一步：执行miniconda3安装程序并指定安装目录为/home/miniconda3 8](#_Toc9416)

[第二步：配置miniconda3环境变量 9](#_Toc13788)

[第三步：查看是否安装成功 9](#_Toc3987)

[第四步：安装依赖包 9](#_Toc19143)

[第五步：解压nlp4.0\_textmatch.tar.gz到miniconda3的envs目录下 9](#_Toc12941)

[5.2 Robot 4.00安装 10](#_Toc6085)

[第一步：上传robot4.00\_20250731.tar安装包到/home/robot4.00目录，并解压 10](#_Toc6260)

[第二步：修改内存大小 10](#_Toc13206)

[第三步：赋予执行权限 10](#_Toc16404)

[第四步：查看robot许可证文件 10](#_Toc8348)

[第五步：启动引擎 11](#_Toc19944)

[第六步：查看catalina.out日志，是否有报错 11](#_Toc30340)

[第七步：查看存在robot进程，则部署成功 11](#_Toc21412)

[第八步：查看python进程是否被拉起 11](#_Toc15902)

[5.3 实体提取接口部署（可选） 12](#_Toc9488)

[5.4 情绪识别接口部署（可选） 13](#_Toc14664)

[6 FAQ 15](#_Toc11498)

[6.1 若服务器性能不够，可以把python程序和java程序放在不同的服务器 15](#_Toc24272)

[方法一：不经过nginx 15](#_Toc2472)

[方法二：经过nginx 18](#_Toc21172)

[6.2 文本匹配缓存写入文件时间、缓存最大句子数修改 22](#_Toc14658)

[6.3 启动robot失败 23](#_Toc9525)

[6.4 robot3.0升级到robot4.0 23](#_Toc5239)

[第一步：停掉robot3.0 24](#_Toc14219)

[第二步：安装依赖包 24](#_Toc7474)

[第三步：拷贝nlp4.0目录 24](#_Toc23364)

[第四步：备份robot-core.jar 24](#_Toc234)

[第五步：配置robot.properties 24](#_Toc10644)

[第六步：修改run.sh 25](#_Toc30519)

[第七步：启动robot3.0 25](#_Toc14873)

[6.3使用普通用户启动robot 26](#_Toc9724)

[6.3.1创建普通用户 26](#_Toc23788)

[6.3.2 依赖包安装 26](#_Toc16325)

[6.3.3 Robot安装 28](#_Toc23019)

[6.5 数据库为mysql5.7版本 30](#_Toc30296)

[6.6 启动robot提示license过期 31](#_Toc1952)

# 声明

## 关于版权

广州云趣信息科技有限公司版权所有，受中国及其他国家商业法律及版权法保护，保留一切权利。未经本公司书面许可，任何单位和个人不得擅自摘抄、复制本书内容的部分或全部，并不得以任何形式传播。

我们尽力保证本文档的完整性和准确性，但不排除文档中有错误存在。广州云趣信息科技有限公司有权在不通知您的情况下更改此文档。

、“YQ”、“yunqu”均为广州云趣信息科技有限公司的商标。对于本手册中出现的其它公司的商标、产品标识及商品名称，由各自权利人拥有。

除非另有约定，本手册仅作为使用指导，本手册中的所有陈述、信息和建议不构成任何明示或暗示的担保。如需要获取最新手册，请登录http://www.yunqu-info.com。

## 阅读约定

|  |  |
| --- | --- |
| **格式** | **意义** |
| **加粗字体** | 手册中，所有加粗的正文字体均表示关键字段。 |
| **红色字体** | 手册中，所有红色显示字体，均表示需要重点、需要特别关注的字段。 |
| **< >** | 手册中，所有带“< >”均表示按钮名，如“单击<确定> 按钮”。 |
| **[ ]** | 带方括号“[ ]”表示窗口名、菜单名和数据表，如“弹出[新建用户]窗口”。 |
| **/** | 多级菜单及并行选项用“/”隔开。如[文件/新建/文件夹]多级菜单表示[文件]菜单下的[新建]子菜单下的[文件夹]菜单项。 |

# 编写目的

用于指导工程、测试人员安装部署robot、进行简单的自检，以及常见问题的定位及解决。

# 部署前准备

|  |  |
| --- | --- |
| 操作系统 | 操作系统为CentOS 7.6 |
| 服务器配置 | 1、8核16G以上且支持avx/avx2（CPU的指令集）  使用cat /proc/cpuinfo 查看cpu指令集  2、机器人并发量大，建议服务器配置32G内存；如果是SAAS服务，建议服务器配置32G内存。  3、ARM架构的服务器配置建议为8C16G \* 1.5 |
| \* 环境依赖 | Conda 23.5.2 |
| \* Robot4.0内存大小 | Robot的java内存分配建议比服务器1/2内存少一点，例如：服务器内存16G，那么Robot的java内存配置7G |
| 端口开放 | 8080 |

# 安装包简介

* **版本服务器下载安装包**：

[http://version.yunqu-info.com/version/4#--产品项目/4.5#--智能化产品安装包/智能AI产品业务模块/robot/robot-V4](http://version.yunqu-info.com/version/4" \l "--产品项目/4.5)



* **安装包列表如下：**

|  |  |  |
| --- | --- | --- |
| **依赖包解压robot4.00\_dependencypackage.zip获得：** | | |
| **依赖包** | **安装服务器** | **安装包描述** |
| conda\_pack-0.7.1-py2.py3-none-any.whl | 引擎服务器 | Conda依赖包 |
| Miniconda3-latest-Linux-x86\_64.sh | 引擎服务器 | Miniconda3安装包(X86) |
| nlp4.0\_textmatch.tar.gz | 引擎服务器 | 文本匹配conda虚拟环境安装包 |
| nlp4.0\_lexical\_analysis.tar.gz（可选） | 引擎服务器 | 实体提取conda虚拟环境安装包 |
| lexical\_analysis.tar.gz（可选） | 引擎服务器 | 实体提取接口安装包 |
| classifier\_emotion.tar.gz（可选） | 引擎服务器 | 情绪识别接口安装包 |
| nlpnet3.tar.gz（可选） | 引擎服务器 | 情绪识别conda虚拟环境安装包 |
| **安装包** | **安装服务器** | **安装包描述** |
| robot4.00\_20250731.tar | 引擎服务器 | NLP引擎安装包 |

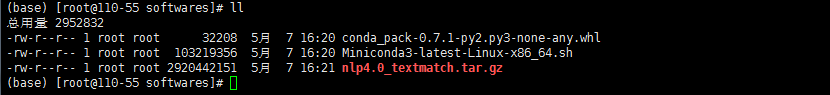
注意：实体提取、情绪识别相关的安装包在SVN的如下路径：[http://version.yunqu-info.com/version/4#--产品项目/4.5#--智能化产品安装包/智能AI产品业务模块/robot/robot-V4/实体提取和情绪识](http://version.yunqu-info.com/version/4" \l "--产品项目/4.5)

# 安装步骤

## 依赖包安装

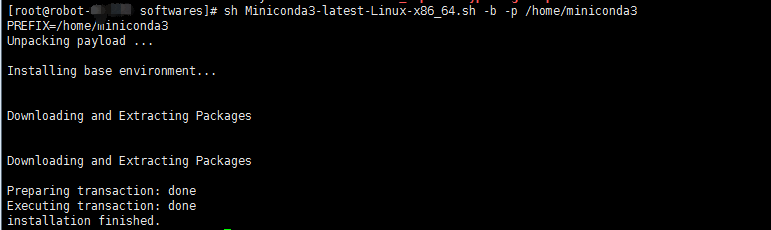
将依赖包都上传到任意自定义路径下，例如该文档中为/home/softwares

该章节所有安装步骤都在依赖包所在路径/home/softwares下执行



**第一步：执行miniconda3安装程序并指定安装目录为/home/miniconda3**

# sh Miniconda3-latest-Linux-x86\_64.sh -b -p /home/miniconda3



**第二步：配置miniconda3环境变量**

在~/.bashrc文件添加如下内容：

# export PATH="/home/miniconda3/bin:$PATH"

**注意：若第一步中的安装目录修改了，这步的路径也要和安装目录一致。**例如第一步的安装目录为/data/miniconda3，则在~/.bashrc文件添加的内容为：export PATH="/data/miniconda3/bin:$PATH"

然后执行如下命令使环境变量生效：

# source ~/.bashrc

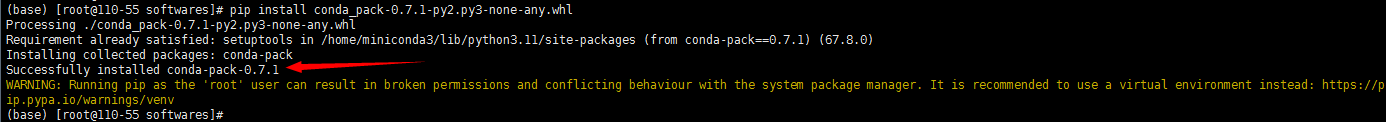
**第三步：查看是否安装成功**

# conda --version



**第四步：安装依赖包**

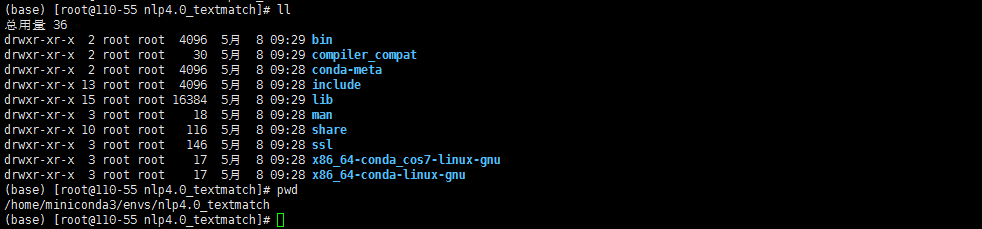
# pip install conda\_pack-0.7.1-py2.py3-none-any.whl



**第五步：解压nlp4.0\_textmatch.tar.gz到miniconda3的envs目录下**

# mkdir /home/miniconda3/envs/nlp4.0\_textmatch

# tar -xzvf nlp4.0\_textmatch.tar.gz -C /home/miniconda3/envs/nlp4.0\_textmatch



## Robot 4.00安装

**第一步：上传robot4.00\_20250731.tar安装包到/home/robot4.00目录，并解压**

# tar -xvf robot4.00\_20250731.tar

**第二步：修改内存大小**

**根据现场实际情况配置，建议比服务器1/2内存少一点点**

# cd /home/robot4.00

# vi run.sh

如果服务器内存16G，可将文件中第一行的两个数字4096分别改成7168。

**第三步：赋予执行权限**

# cd /home/robot4.00

# chmod 777 \*.sh

# cd /home/robot4.00/bin

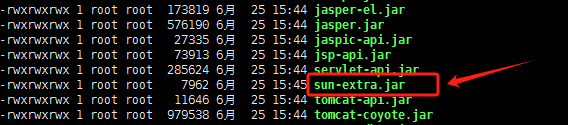
# chmod 777 \*.sh

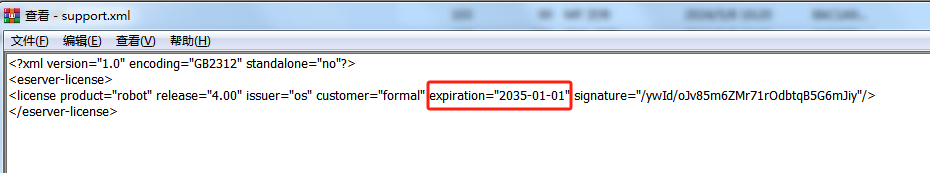
# chmod 777 /home/robot4.00/jdk/jre/bin/java

**第四步：查看robot许可证文件**

# cd /home/robot4.00/lib

打开sun-extra.jar文件下META-INF目录，查看support.xml文件





expiration配置日期为license到期时间（到期后robot无法使用），如需要修改该日期，需要找支持部申请授权，将新的授权文件进行替换。

**第五步：启动引擎**

# cd /home/robot4.00

# ./run\_nohup.sh

按回车键即可



**第六步：查看catalina.out日志，是否有报错**

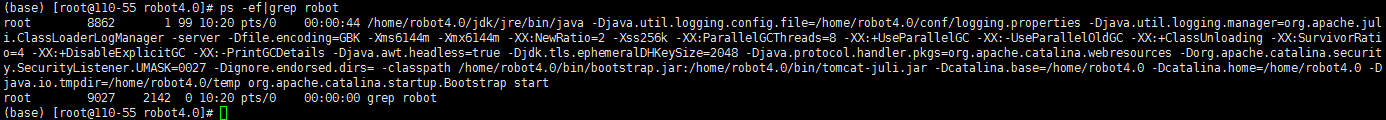
# cd /home/robot4.00/logs

# cat catalina.out | less

按q键退出日志

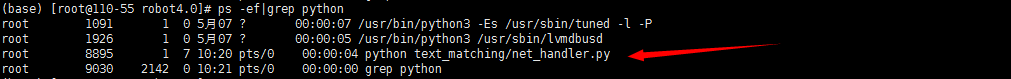
**第七步：查看存在robot进程，则部署成功**

# ps -ef | grep robot



**第八步：查看python进程是否被拉起**

# ps -ef | grep python



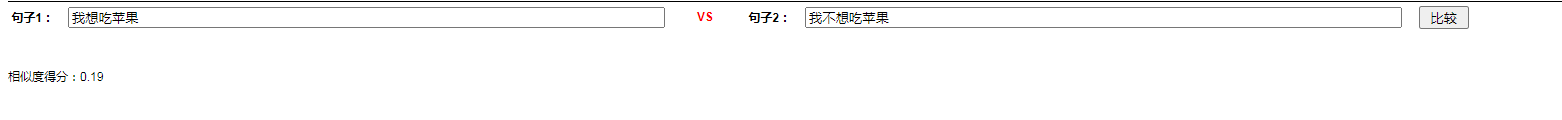
查看python默认端口**8291**是否正常

# netstat -anp |grep 8291



若进程和端口均正常后，可访问NLP的文本匹配页面：

http://IP:端口/robotengine/similarity/query.jsp，IP 为 NLP 引擎所在宿主机的 IP，端口为 NLP 配置的端口（端口在robot应用conf目录下的server.xml配置，默认为8080）。若能正常访问（如下图），则表示Robot安装正常。



## 实体提取接口部署（可选）

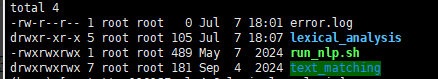
**若项目需要实体提取接口才需要部署。**

1. 把安装包lexical\_analysis.tar.gz解压到robot的nlp4.0目录下

# cd /home/softwares

# tar -zxvf lexical\_analysis.tar.gz -C /home/robot4.00/nlp4.0/

解压后，robot的nlp4.0目录下会有lexical\_analysis目录。



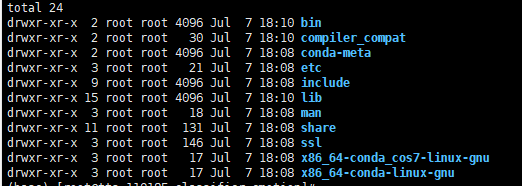
1. 把conda虚拟环境安装包nlp4.0\_lexical\_analysis.tar.gz解压到miniconda3的envs目录下

# mkdir -p /home/miniconda3/envs/nlp4.0\_lexical\_analysis

# cd /home/softwares

# tar -zxvf nlp4.0\_lexical\_analysis.tar.gz -C /home/miniconda3/envs/nlp4.0\_lexical\_analysis

解压后，miniconda3的envs/nlp4.0\_lexical\_analysis目录下会有如下文件。

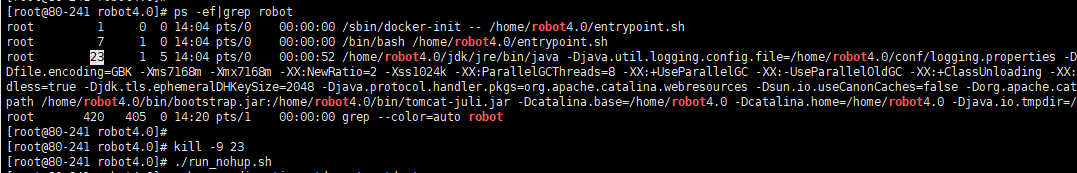


1. 重启robot和python程序

kill掉robot4.0的进程，然后执行

# kill -9 [robot4.0进程号]

# cd /home/robot4.00 && ./run\_nohup.sh

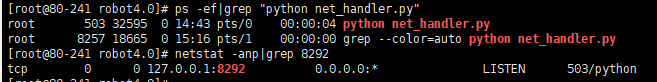


检查python进程以及端口8292是否起来

# ps -ef|grep "python net\_handler.py"

# netstat -anp|grep 8292

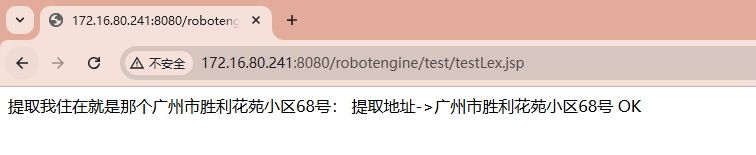
若有如下打印，说明实体提取接口已启动完成。



1. 验证

若进程和端口均正常后，可访问NLP的实体提取页面：

http://IP:端口/robotengine/test/testLex.jsp，IP 为 NLP 引擎所在宿主机的 IP，端口为 NLP 配置的端口（端口在robot应用conf目录下的server.xml配置，默认为8080）。若能正常访问（如下图），则表示实体提取接口正常。



## 情绪识别接口部署（可选）

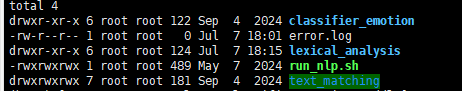
**若项目需要情绪识别接口才需要部署。**

1. 把安装包classifier\_emotion.tar.gz解压到robot的nlp4.0目录下

# cd /home/softwares

# tar -zxvf classifier\_emotion.tar.gz -C /home/robot4.00/nlp4.0/

解压后，robot的nlp4.0目录下会有classifier\_emotion目录。



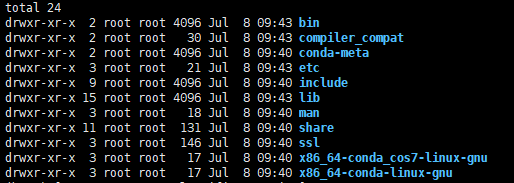
1. 把conda虚拟环境安装包nlpnet3.tar.gz解压到miniconda3的envs目录下

# mkdir -p /home/miniconda3/envs/nlp4.0\_classifier\_emotion

# cd /home/softwares

# tar -zxvf nlpnet3.tar.gz -C /home/miniconda3/envs/nlp4.0\_classifier\_emotion

解压后，miniconda3的envs/nlp4.0\_classifier\_emotion目录下会有如下文件。

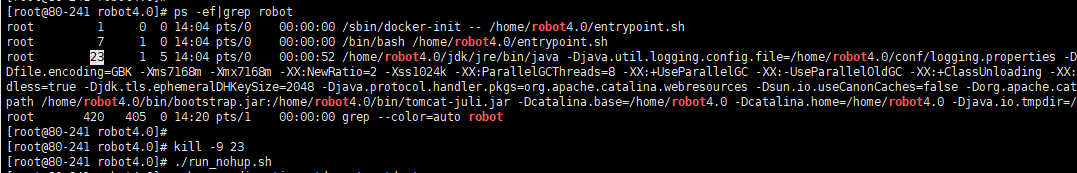


1. 重启robot和python程序

kill掉robot4.0的进程，然后执行

# kill -9 [robot4.0进程号]

# cd /home/robot4.00 && ./run\_nohup.sh

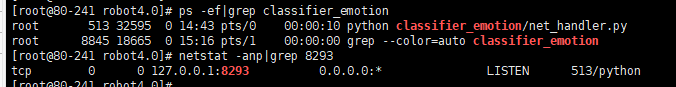


检查python进程以及端口8293是否起来

# ps -ef|grep classifier\_emotion

# netstat -anp|grep 8293

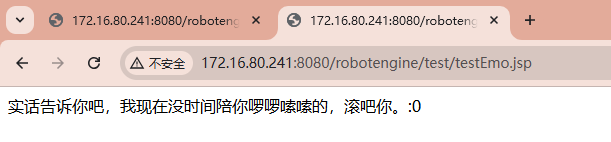
若有如下打印，说明情绪识别接口已启动完成。



1. 验证

若进程和端口均正常后，可访问NLP的情绪识别页面：

http://IP:端口/robotengine/test/testEmo.jsp，IP 为 NLP 引擎所在宿主机的 IP，端口为 NLP 配置的端口（端口在robot应用conf目录下的server.xml配置，默认为8080）。若能正常访问（如下图），则表示情绪识别接口正常。



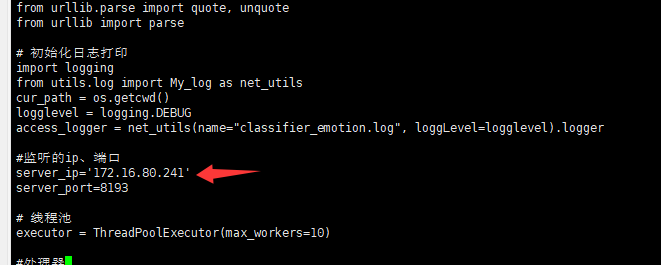
# FAQ

## 若服务器性能不够，可以把python程序和java程序放在不同的服务器

### 方法一：不经过nginx

1. **运行python程序的服务器**

修改robot的nlp4.0目录下text\_matching、lexical\_analysis、classifier\_emotion的net\_handler.py文件（若不需要情绪识别、实体提取的话没有lexical\_analysis、classifier\_emotion目录），将server\_ip改为本机IP

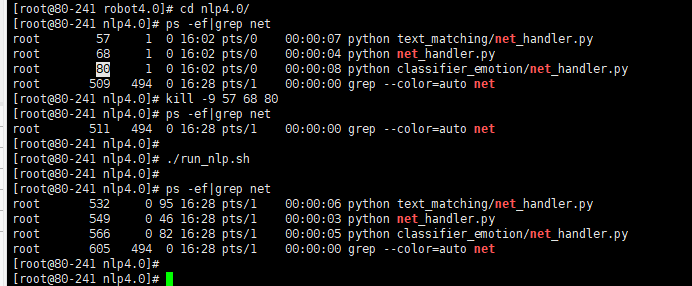


重启python程序

# ps -ef|grep net\_handler.py

# kill -9 [net\_handler.py的三个进程ID]

# cd /home/robot4.00/nlp4.0 && ./run\_nlp.sh

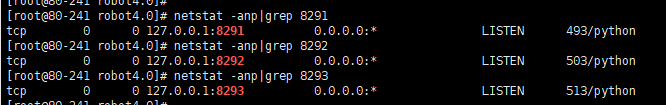


查看text\_matching、lexical\_analysis、classifier\_emotion配置的server\_port是否起来。

# netstat -anp|grep [text\_matching的server\_port]

# netstat -anp|grep [lexical\_analysis的server\_port]

# netstat -anp|grep [classifier\_emotion的server\_port]

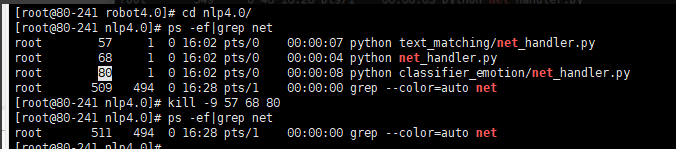


1. **运行java程序的服务器**

先kill掉net\_handler.py相关程序

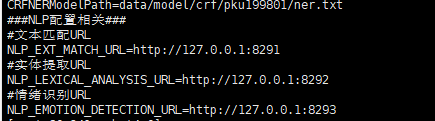
# ps -ef|grep net\_handler.py

# kill -9 [net\_handler.py的进程ID]



然后修改robot的resource/config/robot.properties文件，配置说明：

* NLP\_EXT\_MATCH\_URL：IP为python程序所在服务器IP，端口为python程序所在服务器robot的nlp4.0目录下text\_matching/net\_handler.py文件配置的server\_port（默认为8291）
* NLP\_LEXICAL\_ANALYSIS\_URL：IP为python程序所在服务器IP，端口为python程序所在服务器robot的nlp4.0目录下lexical\_analysis/net\_handler.py文件配置的server\_port（默认为8292）
* NLP\_EMOTION\_DETECTION\_URL：IP为python程序所在服务器IP，端口为python程序所在服务器robot的nlp4.0目录下classifier\_emotion/net\_handler.py文件配置的server\_port（默认为8293）

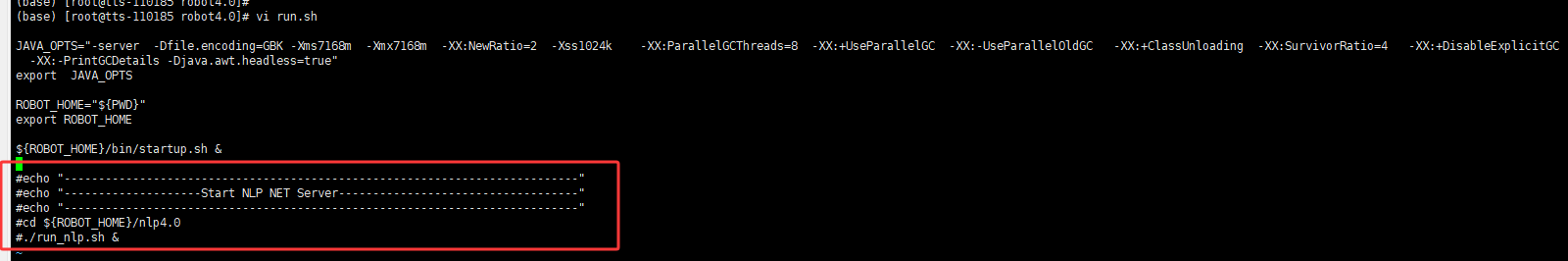


重启java进程

# ps -ef|grep robot

# kill -9 [robot的进程ID]

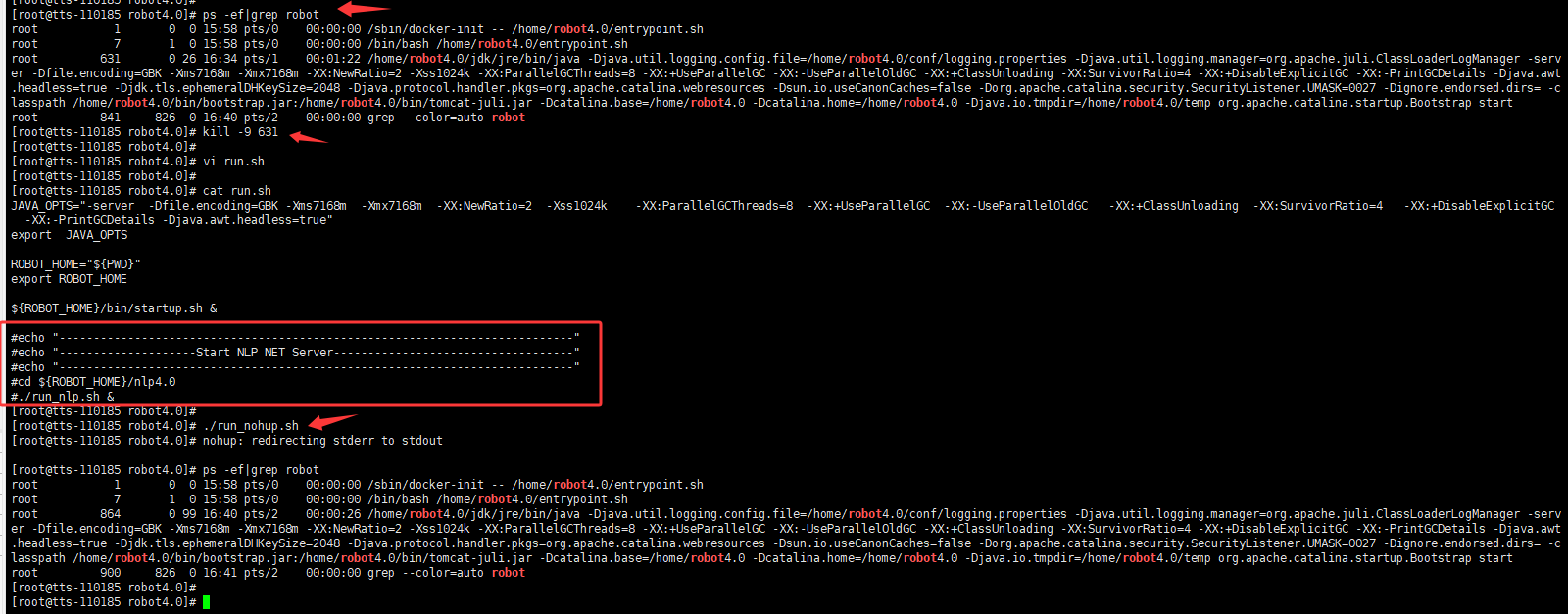
在robot的run.sh文件注释掉如下内容：



然后执行run\_nohup.sh

# cd /home/robot4.00 && ./run\_nohup.sh

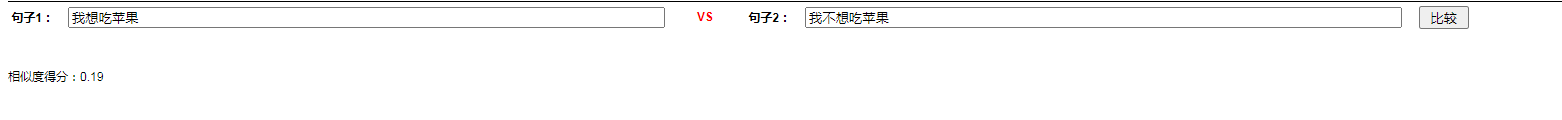
# ps -ef|grep robot



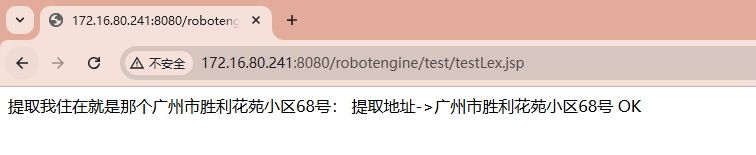
1. 验证：

IP 为java程序所在服务器的 IP，端口为java程序所在服务器robot应用conf目录下的server.xml配置，默认为8080）

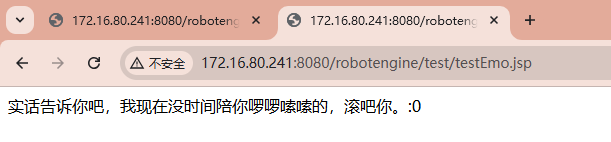
访问NLP的文本匹配测试页面：http://IP:端口/robotengine/similarity/query.jsp



访问NLP的实体提取测试页面：http://IP:端口/robotengine/test/testLex.jsp



访问NLP的情绪识别测试页面：http://IP:端口/robotengine/test/testEmo.jsp



若都能正常访问，则表示robot的功能正常。

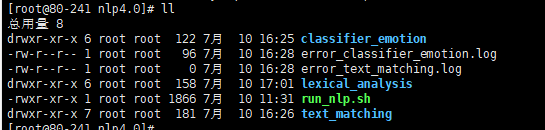
### 方法二：经过nginx

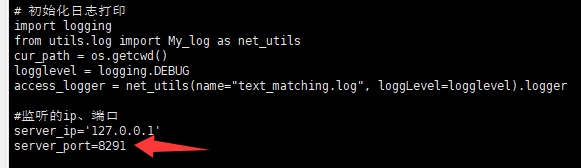
1. **在python程序所在服务器启动nginx**

nginx.conf需要新增的配置如下：

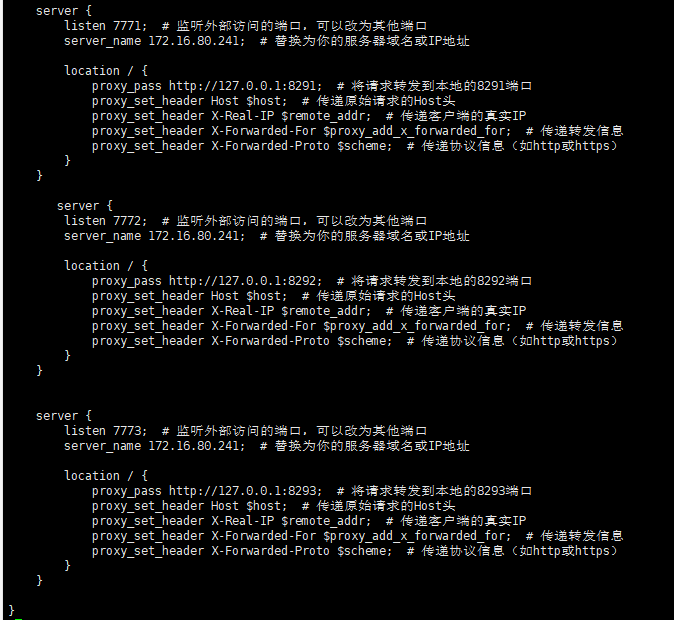
|  |
| --- |
| server {  listen **7771**; # 监听外部访问的端口，可以改为其他端口  server\_name **172.16.80.241**; # 替换为你的服务器域名或IP地址  location / {  proxy\_pass http://127.0.0.1:**8291**; # text\_matching配置的server\_port  proxy\_set\_header Host $host;  proxy\_set\_header X-Real-IP $remote\_addr;  proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;  proxy\_set\_header X-Forwarded-Proto $scheme; # 传递协议信息  }  }    server {  listen **7772**; # 监听外部访问的端口，可以改为其他端口  server\_name **172.16.80.241**; # 替换为你的服务器域名或IP地址  location / {  proxy\_pass http://127.0.0.1:**8292**; # lexical\_analysis配置的server\_port  proxy\_set\_header Host $host;  proxy\_set\_header X-Real-IP $remote\_addr;  proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;  proxy\_set\_header X-Forwarded-Proto $scheme;  }  }  server {  listen **7773**; # 监听外部访问的端口，可以改为其他端口  server\_name **172.16.80.241**; # 替换为你的服务器域名或IP地址  location / {  proxy\_pass http://127.0.0.1:**8293**; # classifier\_emotion配置的server\_port  proxy\_set\_header Host $host;  proxy\_set\_header X-Real-IP $remote\_addr;  proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;  proxy\_set\_header X-Forwarded-Proto $scheme;  }  } |

text\_matching、lexical\_analysis、classifier\_emotion配置的server\_port可以通过robot应用的nlp4.0目录的text\_matching、lexical\_analysis、classifier\_emotion下的net\_handler.py文件查看。



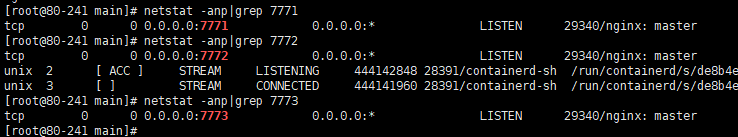


nginx.conf配置截图：



重启nginx，重启完成后查看nginx监听的端口是否都起来了。

# netstat -anp|grep [nginx监听的端口]

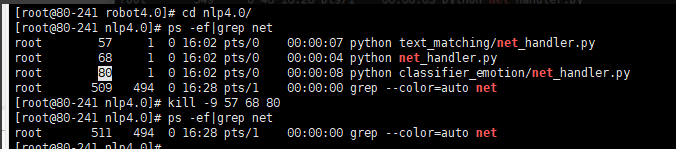


1. **运行java程序的服务器**

先kill掉net\_handler.py相关程序

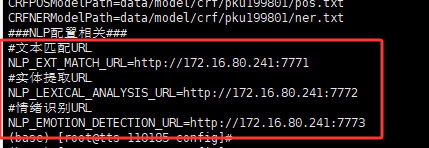
# ps -ef|grep net\_handler.py

# kill -9 [net\_handler.py的进程ID]



然后修改robot的resource/config/robot.properties文件，配置说明：

* NLP\_EXT\_MATCH\_URL：IP为nginx所在服务器IP，端口为nginx配置的text\_matching监听端口
* NLP\_LEXICAL\_ANALYSIS\_URL：IP为nginx所在服务器IP，端口为nginx配置的lexical\_analysis监听端口
* NLP\_EMOTION\_DETECTION\_URL：IP为nginx所在服务器IP，端口为nginx配置的classifier\_emotion监听端口

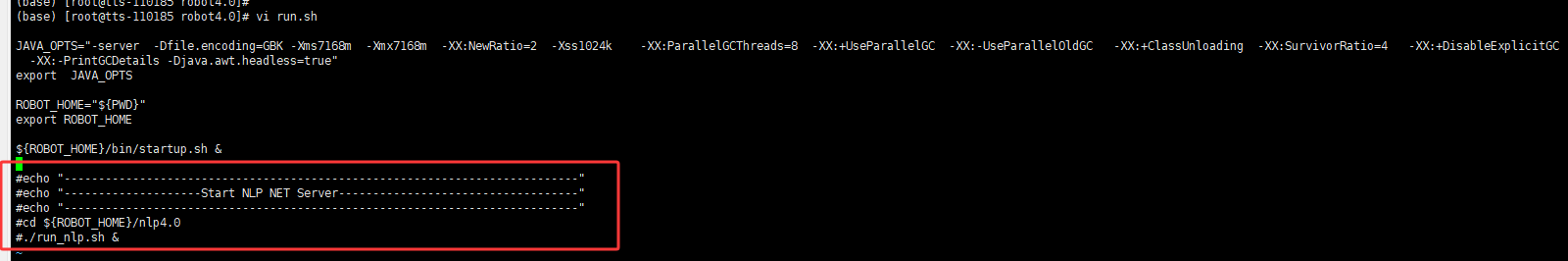


重启java进程

# ps -ef|grep robot

# kill -9 [robot的进程ID]

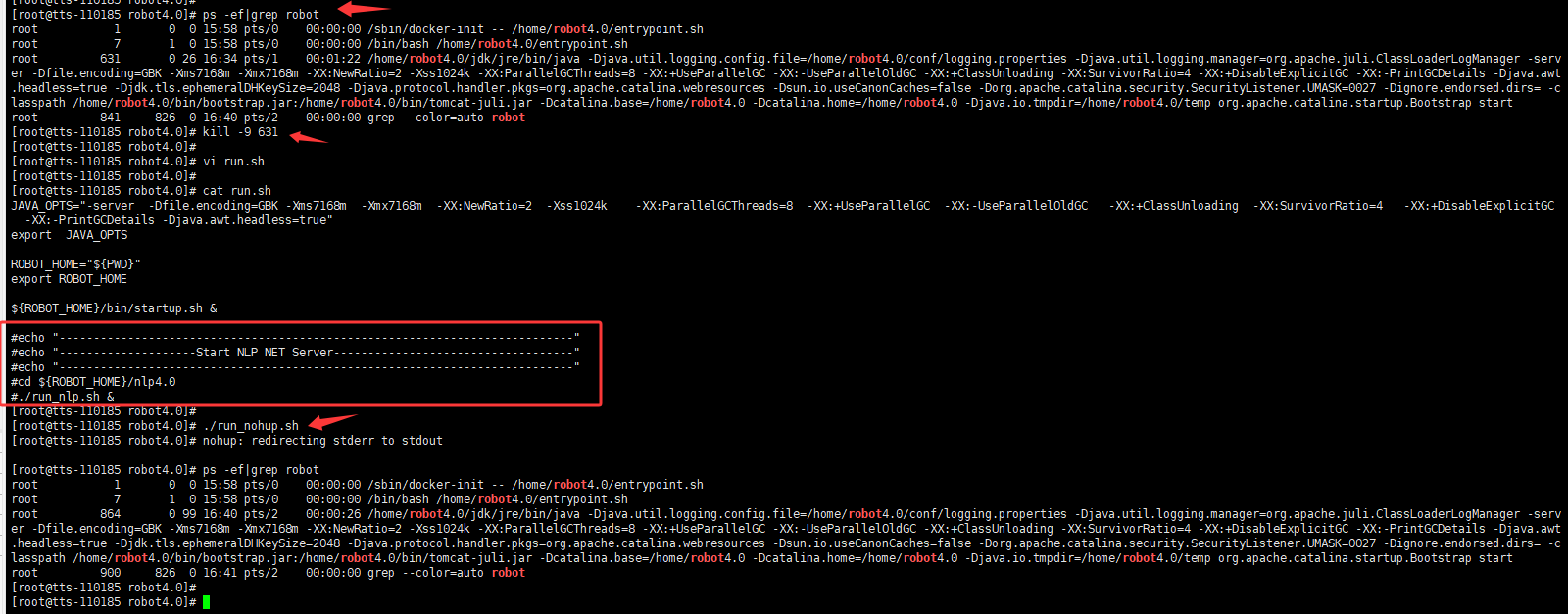
在robot的run.sh文件注释掉如下内容：



然后执行run\_nohup.sh

# cd /home/robot4.00 && ./run\_nohup.sh

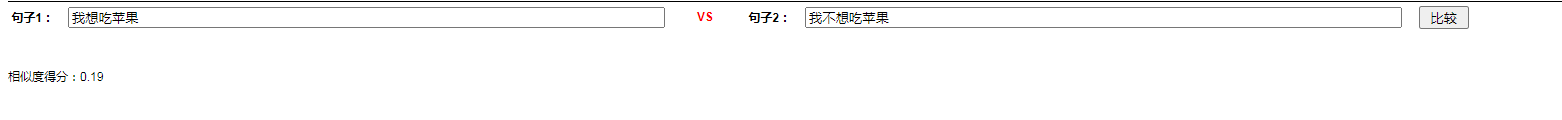
# ps -ef|grep robot



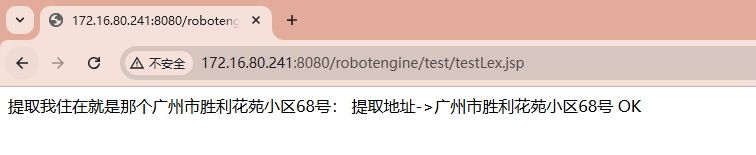
1. 验证：

IP 为java程序所在服务器的 IP，端口为java程序所在服务器robot应用conf目录下的server.xml配置，默认为8080）

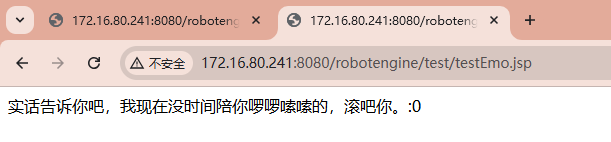
访问NLP的文本匹配测试页面：http://IP:端口/robotengine/similarity/query.jsp



访问NLP的实体提取测试页面：http://IP:端口/robotengine/test/testLex.jsp



访问NLP的情绪识别测试页面：http://IP:端口/robotengine/test/testEmo.jsp

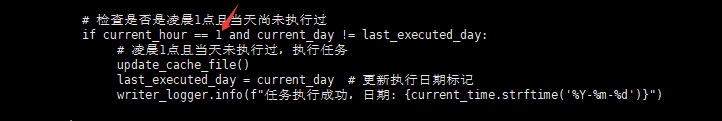


若都能正常访问，则表示robot的功能正常。

## 文本匹配缓存写入文件时间、缓存最大句子数修改

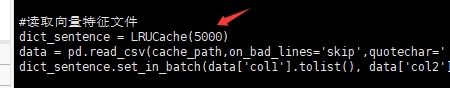
* 文本匹配缓存写入文件时间写入文件时间修改

robot目录的nlp4.0/text\_matching/cache\_file\_writer.py文件，将图中的“1”改成你想设置的时间（24小时制），保存文件然后重启robot



* 文本匹配缓存最大句子数修改

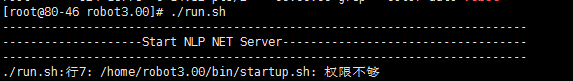
robot目录的nlp4.0/text\_matching/sentence\_transformers\_predict.py文件，将图中的“5000”改成你想设置的缓存的最大句子数，保存文件然后重启robot



## 启动robot失败

第一步：

执行./run.sh查看提示信息：权限不够



第二步：赋予执行权限

cd /home/robot3.00/bin

chmod 777 \*.sh (可按照实际情况给相关脚本权限)

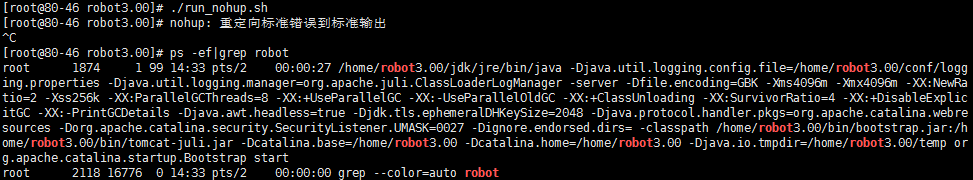
第三步：还未启动成功，查看catalina.out日志



第四步：赋予执行权限

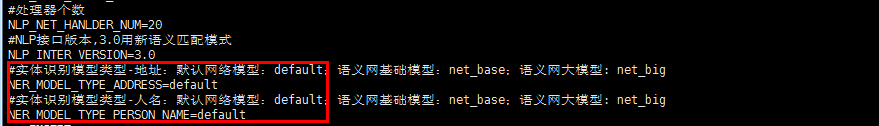
chmod 777 /home/robot3.00/jdk/jre/bin/java

第五步：启动robot



## robot3.0升级到robot4.0

注意：查看robot3.0的robot.properties（/home/robot/resource/config）的NER\_MODEL\_TYPE\_ADDRESS跟NER\_MODEL\_TYPE\_PERSON\_NAME配置值是否为default，如是default则可以进行升级



**第一步：停掉robot3.0**

# pkill -9 java

**第二步：安装依赖包**

按照该文档中的5.1章节进行依赖包安装

**第三步：拷贝nlp4.0目录**

上传robot4.00\_20240530.tar安装包到/home/robot4.00目录，并解压

解压后将nlp4.0目录复制到robot3.0下面

# tar -xvf robot4.00\_20240530.tar

# cp -r nlp4.0/ /home/robot3.0/



**第四步：备份robot-core.jar**

将robot3.0的robot-core.jar进行备份

# cd /home/robot.3.0/resource/lib

# mv robot-core.jar robot-core.jar.bak

将robot4.0的robot-core.jar拷贝至robot3.0

# cp /home/robot4.0/resource/lib/robot-core.jar /home/robot3.0/resource/lib/

**第五步：配置robot.properties**

编辑robot3.0的robot.properties

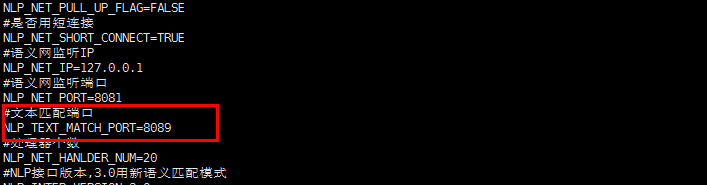
# cd /home/robot3.0/resource/config/

# vi robot.properties

将robot4.0的robot.properties中以下两句添加进去

#文本匹配端口

NLP\_TEXT\_MATCH\_PORT=8089



**第六步：修改run.sh**

编辑robot3.0的run.sh脚本

# vi /home/robot3.0/run.sh

将robot3.0的run.sh脚本中以下几行删除掉

pkill -9 python3

cd ${ROBOT\_HOME}/nlp\_net

python3 net\_handler\_short.py &

将robot4.0的run.sh脚本中的以下几行增加至robot3.0的run.sh中

cd ${ROBOT\_HOME}/nlp4.0

./run\_nlp.sh &

**第七步：启动robot3.0**

# cd /home/robot3.0

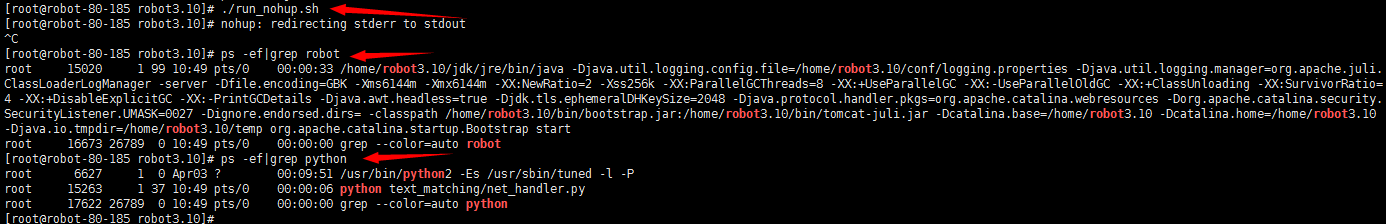
# ./run\_nohup.sh

按回车键即可

查看robot以及python进程

# ps -ef | grep robot

# ps -ef | grep python

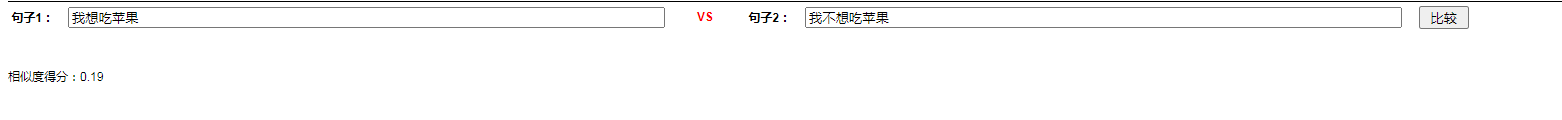


查看python默认端口8291是否正常（robot4.0版本python默认端口由8081改为8291）



若进程和端口均正常后，可访问NLP的数据查询页面：

http://IP:端口/robotengine/similarity/query.jsp，IP 为 NLP 引擎所在宿主机的 IP，端口为 NLP 配置的端口（端口在robot应用conf目录下的server.xml配置，默认为8080）。若能正常访问（如下图），则表示Robot安装正常。



## 6.3使用普通用户启动robot

**6.3.1创建普通用户**

(1) 若启动robot的用户已存在则跳过该小节；

(2) 后续步骤均以 yq 用户作为要启动robot的用户进行说明，若有指定的用户名称，则

需将后续步骤中的“yq”替换成你想要的用户名称。

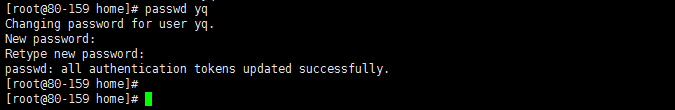
创建用户

# useradd yq

设置用户密码

# passwd yq

需输入两遍密码。



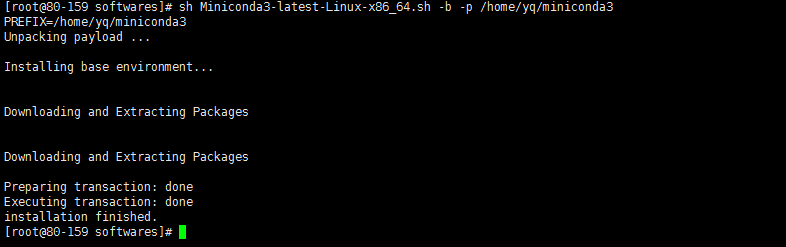
**6.3.2 依赖包安装**

将依赖包都上传到任意自定义路径下，例如该文档中为/home/softwares

该章节所有安装步骤都在依赖包所在路径/home/softwares下执行

**第一步：执行miniconda3安装程序并指定安装目录为/home/yq/miniconda3**

# sh Miniconda3-latest-Linux-x86\_64.sh -b -p /home/yq/miniconda3



**第二步：设置文件目录权限**

# chown -R yq:yq /home/yq

# chmod 777 /home

# chmod 777 /home/yq/.bashrc

**第三步：配置miniconda3环境变量**

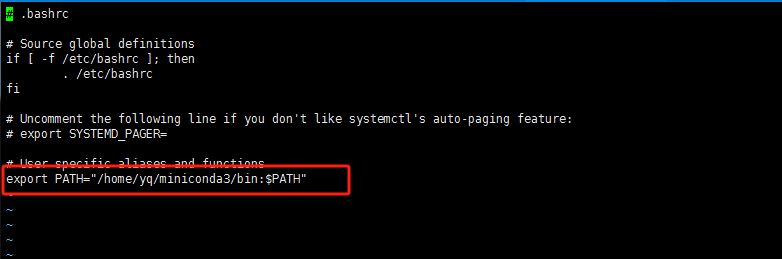
切换到普通用户设置环境变量

# su - yq

# vi /home/yq/.bashrc

添加以下命令：

export PATH="/home/yq/miniconda3/bin:$PATH"



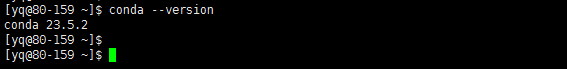
按：输入 wq 保存文件。

执行如下命令使配置立即生效。

# source /home/yq/.bashrc

**第四步：查看是否安装成功**

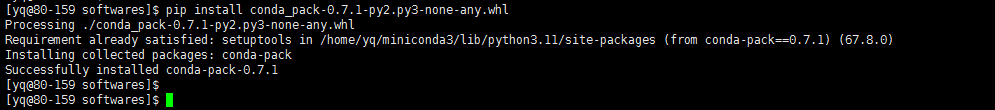
# conda --version



**第五步：安装依赖包**

在/home/softwares目录下执行安装命令

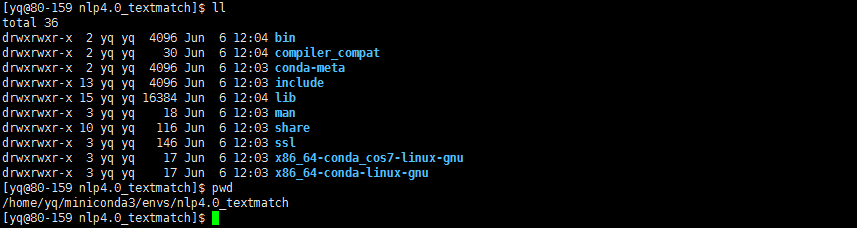
# pip install conda\_pack-0.7.1-py2.py3-none-any.whl



**第六步：解压nlp4.0\_textmatch.tar.gz到miniconda3的envs目录下**

# mkdir /home/yq/miniconda3/envs/nlp4.0\_textmatch

# tar -xzvf nlp4.0\_textmatch.tar.gz -C /home/yq/miniconda3/envs/nlp4.0\_textmatch/



**6.3.3 Robot安装**

**第一步：解压robot4.00\_20240630.tar安装包到/home/yq/robot4.00目录**

# mkdir /home/yq/robot4.00

# tar -xvf robot4.00\_20240630.tar -C /home/yq/robot4.00/

**第二步：修改内存大小**

**根据现场实际情况配置，建议比服务器1/2内存少一点点**

# cd /home/yq/robot4.00

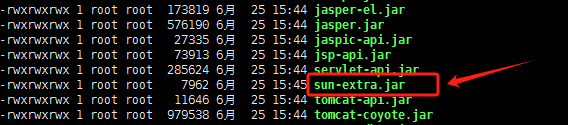
# vi run.sh

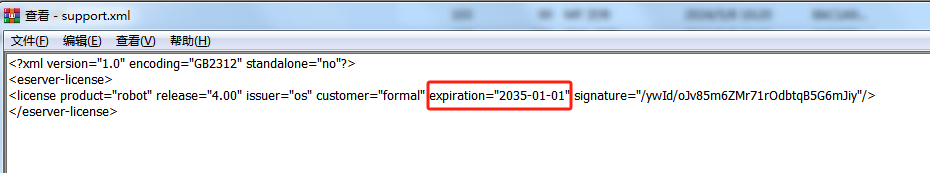
如果服务器内存16G，可将文件中第一行的两个数字4096分别改成7168。

**第三步：查看robot许可证文件**

# cd /home/yq/robot4.00/lib

打开sun-extra.jar文件下META-INF目录，查看support.xml文件





expiration配置日期为license到期时间（到期后robot无法使用），如需要修改该日期，需要找支持部申请授权，将新的授权文件进行替换

**第四步：启动引擎**

# cd /home/yq/robot4.00

# ./run\_nohup.sh

按回车键即可



**第五步：查看catalina.out日志，是否有报错**

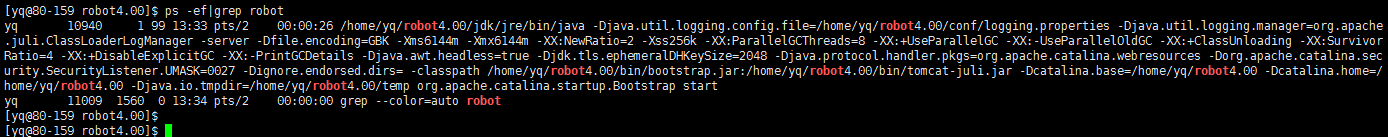
# cd /home/yq/robot4.00/logs

# cat catalina.out | less

按q键退出日志

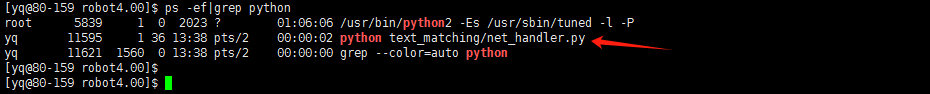
**第六步：查看存在robot进程，则部署成功**

# ps -ef | grep robot



**第七步：查看python进程是否被拉起**

# ps -ef | grep python

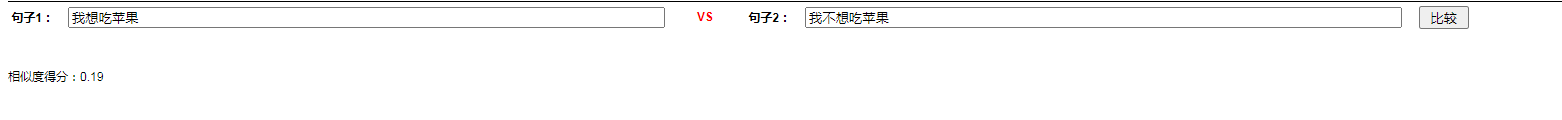


查看python默认端口8291是否正常（robot4.0版本python默认端口由8081改为8291）



若进程和端口均正常后，可访问NLP的数据查询页面：

http://IP:端口/robotengine/similarity/query.jsp，IP 为 NLP 引擎所在宿主机的 IP，端口为 NLP 配置的端口（端口在robot应用conf目录下的server.xml配置，默认为8080）。若能正常访问（如下图），则表示Robot安装正常。

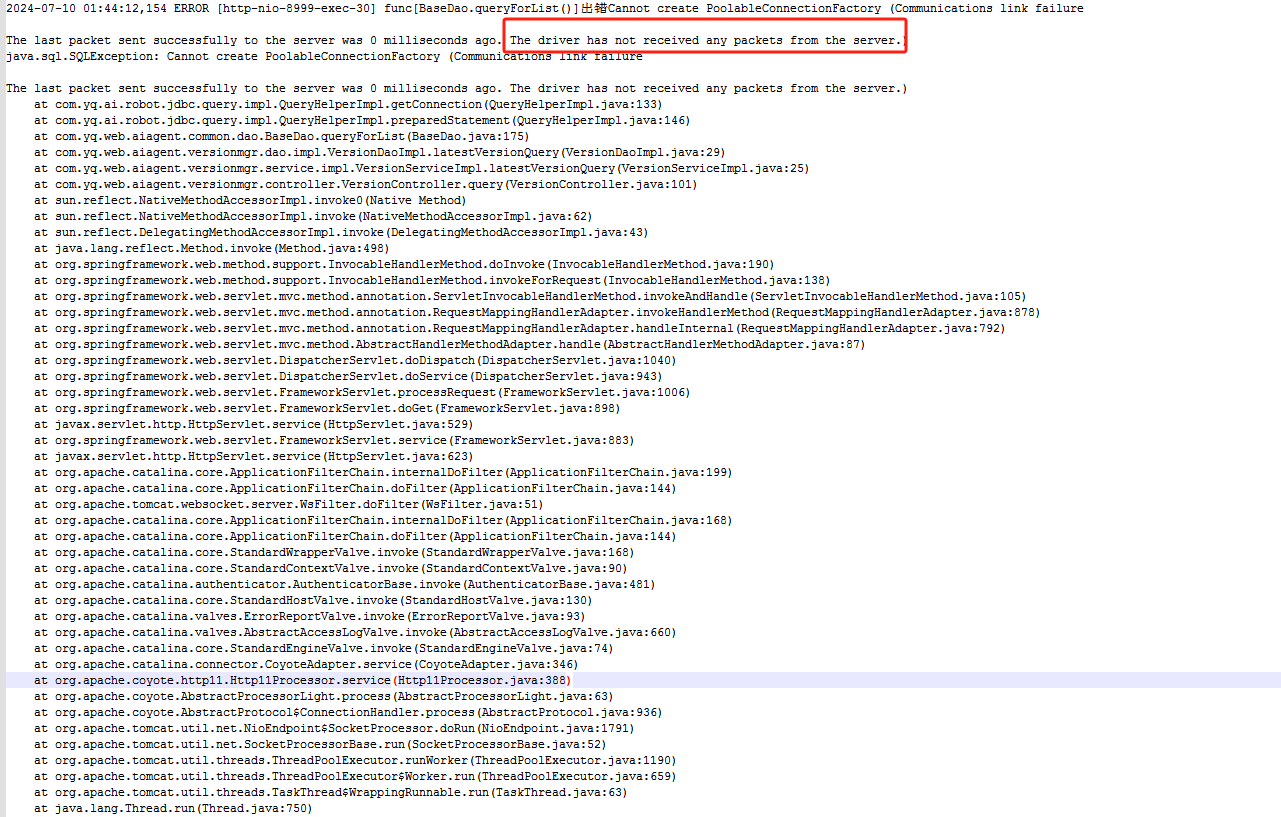


## 数据库为mysql5.7版本

如果数据库为mysql5.7版本，部署robot或者升级robot在产品应用时报错

以语音机器人为例：

机器人训练失败且日志中有以下报错：The driver has not received any packets from the server



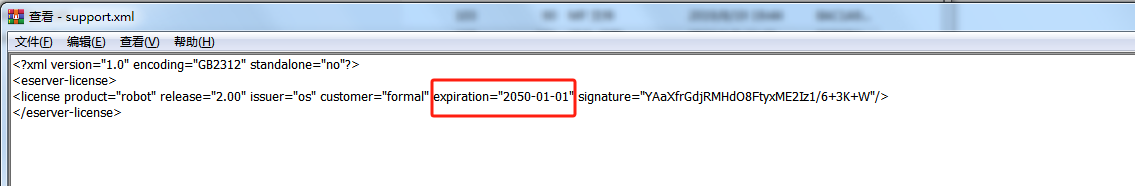
解决方式：可能是驱动不匹配导致连接不到数据库，将/home/robot/resource/lib下的mysql-connector-java-8.0.x.jar备份，再手动将提供的mysql.jar驱动包放至到/home/robot/resource/lib目录下（替换之后需要重启robot）

驱动包在版本库服务器该路径下获取：



## 启动robot提示license过期

查看robot/lib下面的sun-extra.jar，打开该jar包的META-INF目录，查看support.xml文件，expiration配置日期为到期时间



如已经过期，请向支持部申请新的授权文件进行替换