

基于神经网络的图书查询部署文档

本文为基于神经网络的图书查询的部署过程

构建和安装

开发过程基于Ubuntu，使用python3.7，建议使用python3.7虚拟环境

```
$ mkdir your_dir
$ cd your_dir
$ python3.7 -m venv venv
$ source venv/bin/activate
```

安装实验中用到的库（requirements.txt在源码中给出）

```
pip install -r requirements.txt
```

当需要把应用部署到其他地方时，需要构建一个发行文件。当前 Python 的标准发行文件是 *wheel* 格式的，扩展名为 *.whl*。先确保已经安装好 wheel 库：

```
pip install wheel
```

用 Python 运行 *setup.py* 会得到一个命令行工具，以使用构建相关命令。*bdist_wheel* 命令会构建一个 wheel 发行文件。

```
python setup.py bdist_wheel
```

构建的文件为 *dist/flaskr-1.0.0-py3-none-any.whl*。文件名由项目名称、版本号和一些关于项目安装要求的标记组成。

复制这个文件到另一台机器，创建一个新的虚拟环境，然后用 *pip* 安装这个文件。

```
pip install flaskr-1.0.0-py3-none-any.whl
```

pip 会安装项目和相关依赖。

既然这是一个不同的机器，那么需要再次运行 *init-db* 命令，在实例文件夹中创建数据库。

```
export FLASK_APP=flaskr
flask init-db
```

当 Flask 探测到它已被安装（不在编辑模式下），它会与前文不同，使用 *venv/var/flaskr-instance* 作为实例文件夹。

配置密钥

在教程开始的时候给了 *SECRET_KEY* 一个缺省值。在产品中我们应当设置一些随机内容。否则网络攻击者就可以使用公开的 *'dev'* 键来修改会话 *cookie*，或者其他任何使用密钥的东西。

可以使用下面的命令输出一个随机密钥：

```
python -c 'import os; print(os.urandom(16))'

b'_5#y2L"F4Q8z\n\xec]/'
```

在实例文件夹创建一个 *config.py* 文件。工厂会读取这个文件，如果该文件存在的话。提制生成的值到该文件中。

```
venv/var/flaskr-instance/config.py
SECRET_KEY = b'_5#y2L"F4Q8z\n\xec]/'
```

其他必须的配置也可以写入该文件中。Flaskr 只需要 SECRET_KEY 即可。

运行产品服务器

当运行公开服务器而不是进行开发的时候，应当不使用内建的开发服务器（`flask run`）。开发服务器由 Werkzeug 提供，目的是为了更方便开发，但是不够高效、稳定和安全。

替代地，应当选用一个产品级的 WSGI 服务器。例如，使用 Waitress。首先在 虚拟环境中安装它：

```
pip install waitress
```

需要把应用告知 Waitree，但是方式与 `flask run` 那样使用 `FLASK_APP` 不同。需要告知 Waitree 导入并调用应用工厂来得到一个应用对象。

```
waitress-serve --call 'flaskr:create_app'
```

```
Serving on http://47.101.161.57
```