

文章

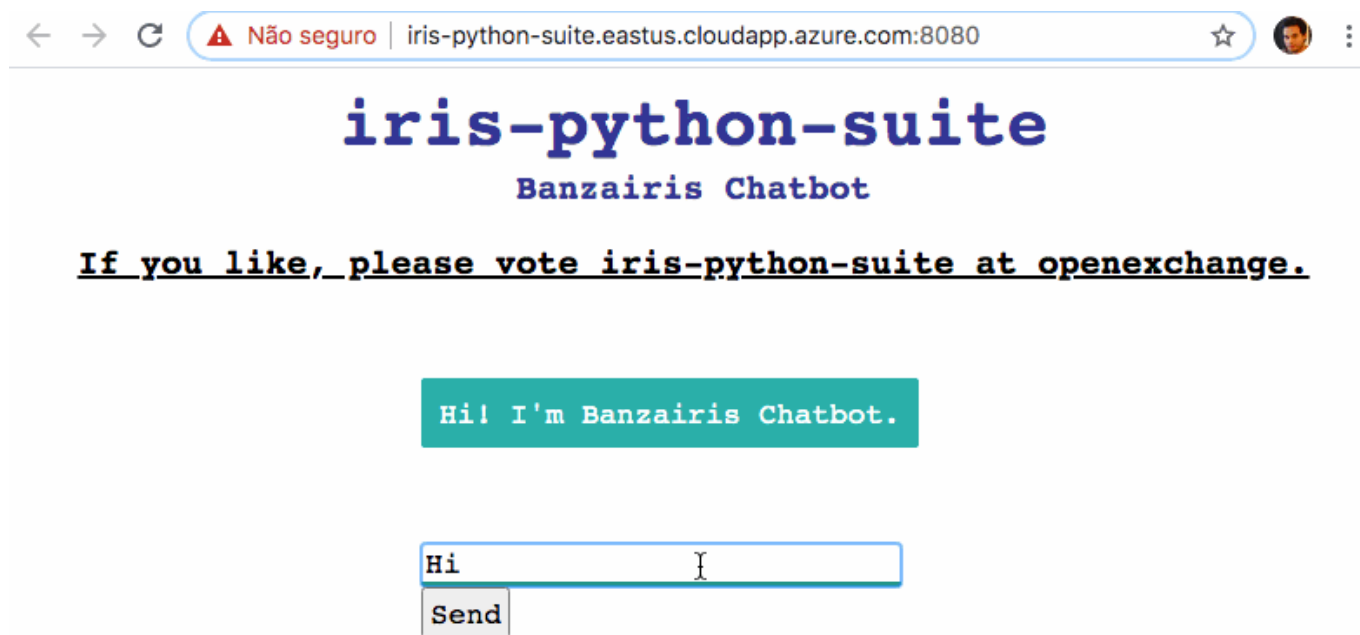
[Hao Ma](#) · 三月 26, 2021



阅读大约需分钟

[Open Exchange](#)

使用 IRIS 和 Python 创建聊天机器人



使用 IRIS 和 Python 创建聊天机器人

本文将展示如何将 InterSystems IRIS 数据库与 Python 集成服务于自然语言处理 (NLP) 的机器学习模型。

为何选择 Python?

随着在世界范围内的广泛采用和使用, Python 拥有了出色的社区, 以及许多加速器 | 库用于部署任何类型的应用。如果您感兴趣请访问 <https://www.python.org/about/apps/>

Iris Globals

我接触到 ^globals 后很熟悉了，它们可以用作快速获取现有数据模型中数据的方法。因此，首先，我将使用 ^globals 存储训练数据和对话以记录聊天机器人的行为。

自然语言处理

自然语言处理或 NLP 是 AI 的一个主题，它创造了机器从我们的语言阅读、理解含义的能力。显然，这并不简单，但是我将展示如何在这个广阔而美丽的领域中迈出您的第一步。

演示 - 来试试吧

我在这里部署了 Chatbot 应用作为演示：<http://iris-python-suite.eastus.cloudapp.azure.com:8080>

工作原理

机器学习

要知道与普通软件开发相比，机器学习具有不同的范式。很难理解的要点是机器学习模型的开发周期。

浅显解释预警

一个标准应用开发周期大概是这样的：

开发代码 -> 测试(使用开发数据) -> 部署(真实用户数据)

机器学习代码本身不具有相同的价值。它会与数据分担责任！而且不是任意数据，是真实数据！因为待执行的最终代码是由开发概念和所用数据合并生成所以机器学习应用周期类似于：

开发(训练)模型 + 真实数据 -> 验证 -> 部署此模型的结果

如何训练模型？

训练模型的方法有很多，每种情况和目标都需要很大的学习曲线。在本例中，我使用的是 [ChatterBot](#) 库，该库封装了一些方法，并提供了训练方法和经过预处理的训练数据，有助于我们关注结果。

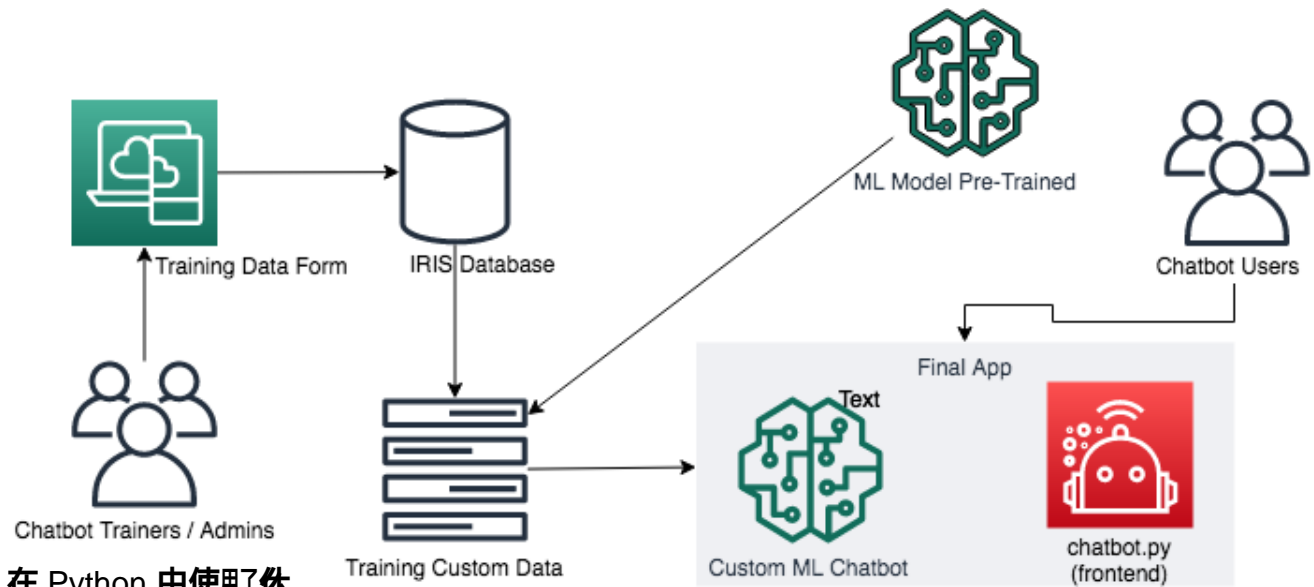
预训练的模型语言和自定义模型

您可以由此开始拥有一个基本的会话聊天机器人。

您还可以创建所有数据来训练您的聊天机器人，全面满足您的需求，但这在短时间内很难完成

在这个项目中，我使用 en_core_web_sm 作为对话的词典，并与可以通过 [表单](#) 创建的自定义训练数据合并

结构



在 Python 中使用了什么

在这个应用环境中，我使用了 Python 3.7 和这些模块：

- PyYAML<=5.0.0
- dash==1.12.0
- dash-bootstrap-components==0.10.1
- dash-core-components==1.10.0
- dash-html-components==1.0.3
- dash-renderer==1.4.1
- dash-table==4.7.0
- plotly==4.7.1
- numpy==1.18.4
- networkx==2.4
- Flask>=1.0.0
- chatterbot>=1.0.0
- chatterbot-corpus>=1.2.0
- SQLAlchemy>=1.2
- ./nativeAPI_wheel/irisnative-1.0.0-cp34-abi3-linux_x86_64.whl

项目结构

本项目具有简单易懂的结构。在主文件夹上，有 3 个最重要的子文件夹：

- ./app：具有全部 **应用代码** 和安装配置。
- ./iris：具有 InterSystems IRIS dockerfile，准备服务于应用。
- ./data：通过一个 **卷**将主机链接到容器环境

应用结构

现在，可以在 ./app 目录翻一些文件：

- chatbot.py：具有 Web 应用实现
- iris_python_suite.py：具有一些加速器的类，通过 IRIS Native API 与 IRIS 数据库和 Python 搭配使用。

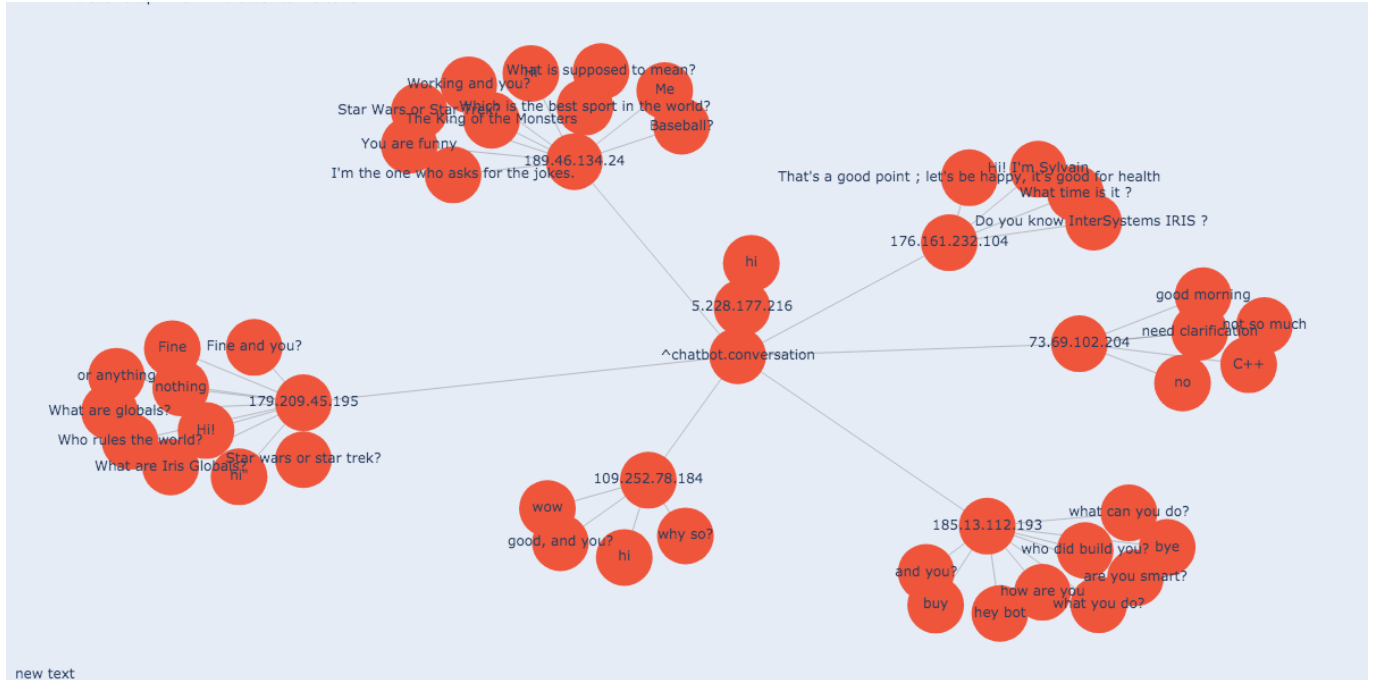
数据库结构

此应用使用 InterSystems IRIS 作为存储库，使用的 globals 包括：

- ^chatbot.training.data : 以问题和答案的形式存储所有自定义训练数据。
- ^chatbot.conversation : 存储所有对话有效负载。
- ^chatbot.training.isupdated : 控制训练管道。

其他解决方案的产品

我没有为所有对话创建报告，但这不是个问题，使用全图可视化即可跟踪对话。



亲自运行应用

先决条件

- git
- docker 和 docker-compose(以及 docker 中更够内存设置, 至少 4GB)
- 在您的环境中访问终端

步骤

使用 docker-compose, 您可以在一个环境中将所有组件和配置轻松转到 iris-python-covid19 文件夹, 键入:

```
$ docker compose build  
$ docker compose up
```

估计转入容器的时间

第一次运行将通过网络链接载图像和依赖项。如果持续时间超过 15 分钟, 那可能是出现了问题, 请随时来这里留言。第一次运行后, 下一次运行会好很多, 只需不到 2 分钟。

如果一切正常

一段时间后, 您可以打开浏览器并转到地址:

训练数据表单

<http://localhost:8050/chatbot-training-data>

聊天机器人

<http://localhost:8080>

您应该查看 IRIS 管理门户

暂时使用的是 USER 命名空间

<http://localhost:9092>

user: _SYSTEM

pass: theansweris42

如果本文对您有帮助或者有您喜欢的内容,请投票:

此应用目前正在参与 Open Exchange 竞赛,您可以在这里给应用投票
(<https://openexchange.intersystems.com/contest/current>)

iris-python-suite 投票

[#AI](#) [#Python](#) [#机器学习](#) [#框架](#) [#InterSystems IRIS](#) [#其他](#)
[在 InterSystems Open Exchange 上检查相关应用程序](#)

源 URL: <https://cn.community.intersystems.com/post/%E4%BD%BF%E7%94%A8-iris-%E5%92%8C-python-%E5%88%9B%E5%BB%BA%E8%81%8A%E5%A4%A9%E6%9C%BA%E5%99%A8%E4%BA%BA>