案例二：基于rasa的多轮对话任务机器人

1.案例目的

• 了解多轮对话机器人的构建技术；

• 掌握rasa多轮对话机器人框架的使用

• 掌握多轮对话机器人构建流程

• 掌握训练和测试对话机器人

2.案例内容

本次案例将带领同学们利用Rasa搭建一个基于犯罪信息知识图谱的多轮智能对话检索机器人。Rasa是一个用于对话的开源机器学习框架。首先了解用户想说的内容（NLU），然后根据上下文信息对其进行适当的操作对话（Core）。

当前对话系统主要包括4类：

• 问答型对话系统

• 任务型对话系统

• 闲聊型问答系统

• 图谱型问答系统



Rasa是一个用于对话的开源机器学习框架。首先了解用户想说的内容（NLU），然后根据上下文信息对其进行适当的操作对话（Core）。

Rasa核心模块

• NLU - 自然语言理解；该模块主要负责对用户输入的文本进行意图识别和实体抽取，理解用户想要做什么以及其中包含的关键信息。

输入：一串自然语言的文本

输出：分词和实体提取意图识别之后的结构化文本，(intent，slot，value): 用一个三元组结构化描述text，比如 “姚明的身高是81吗？” 对应的三元组 （询问身高，hight，81）

• Core - 对话管理

​    对话管理（Dialog Management，DM）控制着人机对话的进程，对话管理通过对自然语言理解模块的输出（即意图、槽位、槽位信息）的处理来进行和用户的交互和反应。针对带有明确意图的用户，在对话管理过程中用户不但可以完善或修正自己需求，管理系统也可以在用户需求不够具体的时候明确或澄清用户的需求，以达到最终的任务完成。

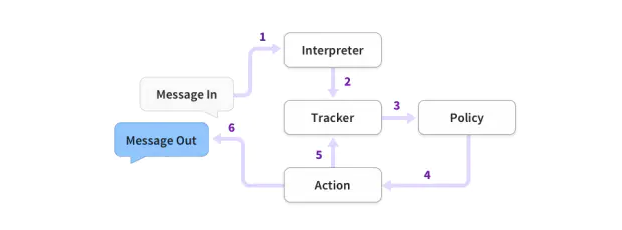
​    对话管理系统主要包括两部分：状态追踪和对话策略。对话策略通常作为对话管理的输出，如对该场景下缺失槽位的反问策略等。以某公司智能产品介绍系统的多伦对话为示例：



• NLG - 自然语言生成

​    基于多伦对话的自然语言生成，通常也采用基于模板、基于语法或模型等。模板与语法主要基于规则的策略，模型可以用如seq2seq等网络生成自然语言。

Rasa对话流程图



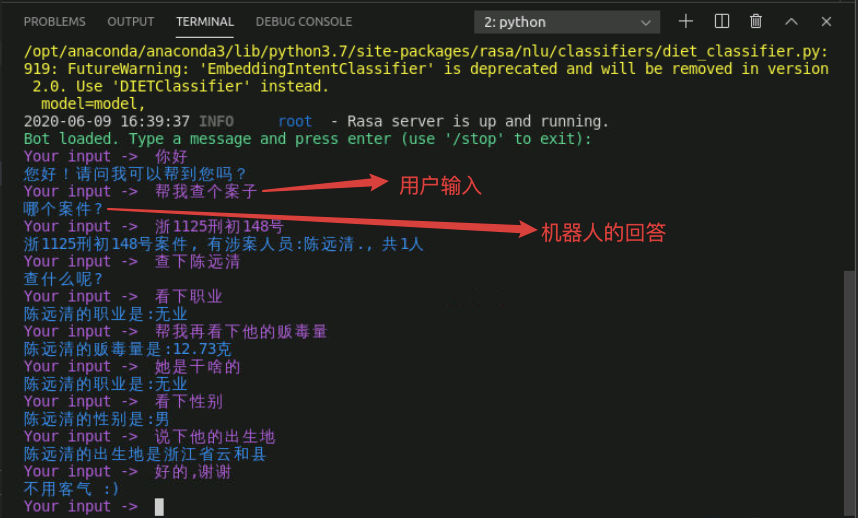
• 1、将用户输入的Message传递到Interpreter(NLU模块)，并将信息打包为一个字典, 该模块负责识别Message中的"意图(intent)“和提取所有"实体”(entity)数据；

• 2、Rasa Core会将Interpreter提取到的意图和识别传给Tracker对象，保持对话状态(Conversation State)；

• 3/4/5、利用Policy记录Tracker对象的当前状态，并选择执行相应的Action，其中，这个Action是被记录在Track对象中的；

• 6、将执行Action返回的结果输出，即完成一次人机交互（一轮对话结束）。

最终，我们搭建的多轮对话任务机器人运行效果如下图所示：



3.案例知识点

• 基于py2Neo和neo4j搭建知识图谱；

• 意图识别；

• 实体识别提取。

• 对话管理；

• rasa框架使用

4.案例时长

共4学时，学时分配如下：

• 知识图谱搭建（1学时）；

• rasa智能对话机器人搭建（3学时）。

5.案例实验环境

• ubutun；

• neo4j

• Pycharm

• Python3.6

• py2neo 4.3.0

• rasa 1.10.1

6.案例分析

1）项目准备

2）知识图谱搭建

3）rasa智能对话机器人搭建

• 编写域文件domain.yml；

• 编写自然语言理解训练文件nlu.md；

• 查看对话管理训练文件stories.md

• 编写对话业务处理脚本文件actions.py

• 编写分词字典文件user\_dict.txt

• 编写配置文件config.yml

• 编辑endpoints.yml文件

4）训练和测试机器人

7.案例实验过程

7.1项目准备

1) 创建项目

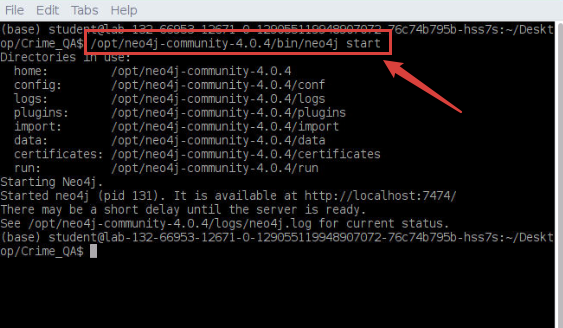
使用Pycharm工具，创建一个 Project，项目名称为“Crime\_QA”。

2) 启动neo4j服务

在打开的命令窗口中输入以下命令，回车执行：

/opt/neo4j-community-4.0.4/bin/neo4j start

该命令为启动neo4j数据库服务的命令，具体启动的路径要根据案例所用电脑中neo4j安装的位置决定，启动成功后终端输出如下信息：

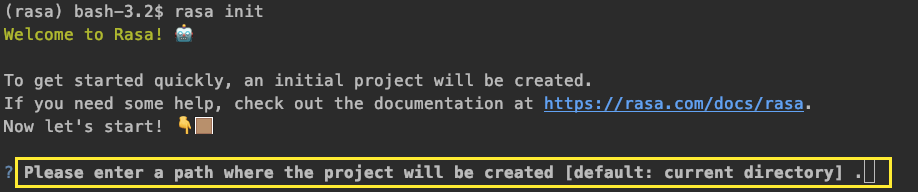


3) 构建rasa框架代码

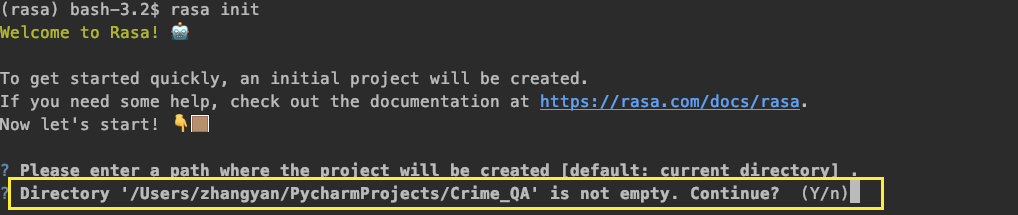
打开pycharm下部的Terminal窗口，输入以下命令：

rasa init

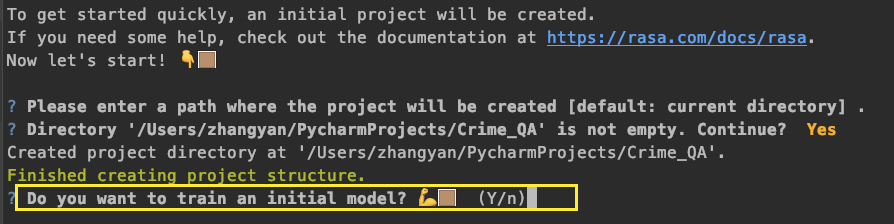
回车执行后，出现一句话要求输入路径，这里直接回车，使用当前路径即可。如下图所示：



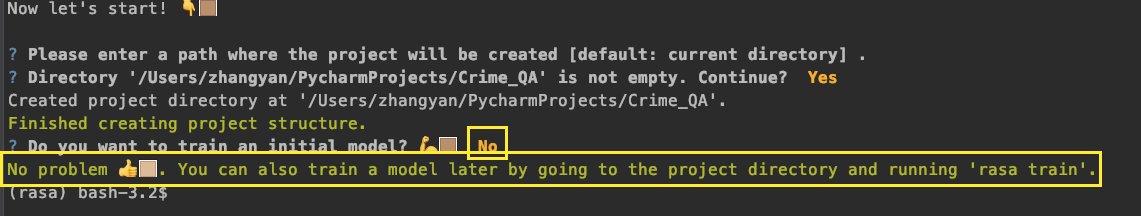
回车执行后，又出现一句话询问是否继续，直接回车继续。如下图所示：



紧接着再次出现一句话询问是否继续，这次需要输入：n。如下图所示：

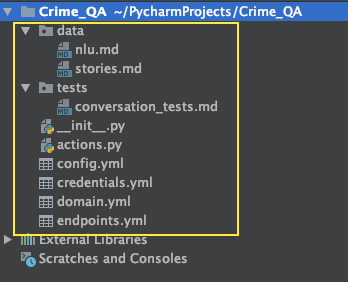


输入n后，命令执行结束，结果如下图所示：



4) 查看rasa框架生成的代码

命令执行成功后，我们可以看到项目目录中自动生成了一些目录和文件，如下图所示：



图中显示的就是rasa框架自动生成的任务型多轮对话机器人项目的初始代码结构，项目名称是：Crime\_QA，相关目录及文件的具体说明如下：

1.data 目录：此目录用于存放训练所需的数据，是上一步中rasa init命令生成的。

• nlu.md：该文件中存放的是自然语言理解（nlu）模块训练所需的数据，目前文件内容是自动生成的，我们将在后续的步骤中进行修改。

• stories.md：该文件中存放的是对话管理（DM）模块训练所需的故事（story），后续需要修改。

2.material 目录：此目录中的内容不是系统生成，而是本案例为同学们准备好的训练数据和相关材料。

• nlu.md：该文件中存放的是本次案例提供的自然语言理解（nlu）模块训练所需的数据。

• stories.md：该文件中存放的是本次案例提供的对话管理（DM）模块训练所需的故事（story）。

3.tests目录：该目录为系统自动生成的测试数据文件，本次案例中不会使用。

4.\_\_init\_\_.py：空文件。

5.actions.py：可以自定义 actions 的代码文件，后面的步骤中需要修改该文件。

6.config.yml：Rasa 的配置文件，该文件需要进行修改。

7.credentials.yml：定义和其他服务连接的一些细节，例如rasa api接口，本次案例中无需编辑。

8.domain.yml：Rasa domain 文件，该文件需要进行编辑。

9.endpoints.yml：和外部消息服务对接的 endpoins 细则，例如 fb messenger，该文件需要简单编辑。

5) 复制文件

将案例提供的数据文件neo4j\_demo.json复制到data目录下，将storie.md文件复制到data目录下替换同名文件，结构如图7所示：

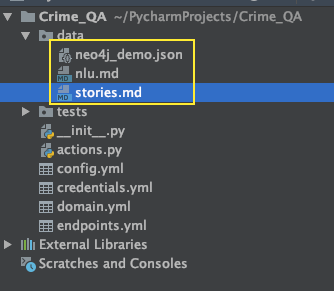


图7 复制数据文件结果

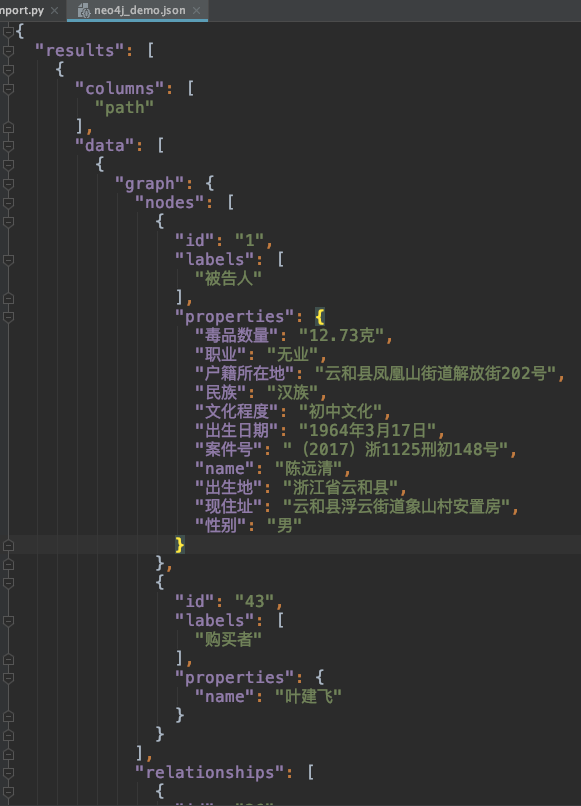
各文件作用如下：

• neo4j\_demo.json：犯罪案件信息数据文件，后续用来写入neo4j数据库构建小型知识图谱

• storie.md：该文件中存放的是本次案例提供的对话管理（DM）模块训练所需的故事（story），由于内容较多，因此这里直接提供。

7.2知识图谱搭建

本节主要任务是基于提供的犯罪案件数据文件neo4j\_demo.json，利用neo4j图数据库搭建小型知识图谱，并基于建立的知识图谱，为后续的智能问答(QA)提供知识服务。neo4j\_demo.json文件的内容如下图所示：



我们需要编写脚本将这份数据解析，并写入neo4j图数据库。

在Crime\_QA项目中新建Python File文件“data\_import”。

1）导入需要使用的包。

• json用来解析json字符串

• Graph,Node, Relationship, NodeMatcher四个类用来操作neo4j图数据库

【代码1】build\_medicalgraph.py

import json

from py2neo import Graph, Node, Relationship, NodeMatcher

2）定义类MyNode。

MyNode类的主要作用是描述一个实体节点，存储该节点的标签（labels）和属性（properties）。类的方法有三个：

• \_\_init\_\_初始化方法：该方法接收一分表示Node节点数据的字典，然后解析该字典提取节点的id、标签labels和属性properties，并将这些信息保存到类的属性中。

• get\_node方法：该方法的作用是返回使用属性中存储的数据构建的Node对象，该对象可以直接存入neo4j数据库。

• \_\_repr\_\_方法：重写的对象方法，用于打印对象。

【代码2】build\_medicalgraph.py

class MyNode(object):

def \_\_init\_\_(self, node\_dict):

self.id = node\_dict.get('id') # 提取id

self.labels = node\_dict.get('labels')#提取标签

self.properties = node\_dict.get('properties')#提取属性

self.properties['id'] = str(node\_dict.get('id'))#提取id并存入属性

def \_\_repr\_\_(self) -> str:

return str(self.\_\_dict\_\_)

def get\_node(self):

return Node(\*\*self.labels, \*\*self.properties)#返回使用数据构造的Node对象

3）定义类MyRelation。

MyRelation类的主要作用是描述一个关系，存储关系的id、关系类型（type）和关系的属性（properties）。类的方法有两个：

• \_\_init\_\_初始化方法：该方法接收一份表示关系数据的字典，然后解析该字典提取关系的相关数据：

• 关系的id

• 关系类型（type）

• 起始节点的id（startNodeId）

• 终止节点的id（endNodeId）

• 关系的属性（properties）

并将这些信息保存到类的属性中。

• \_\_repr\_\_方法：重写的对象方法，用于打印对象。

【代码2】build\_medicalgraph.py

class MyRelation(object):

def \_\_init\_\_(self, relation\_dict):

self.type = relation\_dict.get('type')#提取关系类型

self.startNodeId = str(relation\_dict.get('startNode'))#提取起始节点

self.endNodeId = str(relation\_dict.get('endNode'))#提取终止节点

self.properties = relation\_dict.get('properties')#提取属性

self.properties['id'] = str(relation\_dict.get('id'))#提取id并存入属性

def \_\_repr\_\_(self) -> str:

return str(self.\_\_dict\_\_)

4）读取数据文件并解析。

定义好两个类之后，我们读取数据文件，并提取数据，将数据封装为MyNode对象和MyRelation对象，具体流程如下：

• 读取数据文件neo4j\_demo.json，并使用json加载为字典结构

• 定义用来保存MyNode对象的字典node\_dict和保存MyRelation对象的列表relationships

• for循环遍历数据中的每一个graph

• 遍历所有的nodes，使用数据实例化MyNode对象并放入node\_dict中保存

• 遍历所有的relationships，使用数据实例化MyRelation对象并放入relationships中保存

【代码2】build\_medicalgraph.py

with open('./data/neo4j\_demo.json', 'r', encoding='utf-8') as f:

data = json.loads(f.read())#数据转换成字典

data = data.get('results')[0].get('data')#获取数据

node\_dict = {} # 节点字典

relationships= [] # 关系列表

# 获取所有节点和关系

for \_graph in data:

\_graph = \_graph.get('graph')

for node in \_graph.get('nodes'):#遍历每一个node信息

n = MyNode(node)

node\_dict[str(n.id)] = n#返回节点对象存入字典

for relation in \_graph.get('relationships'):#遍历每一个relationship信息

relationships.append(MyRelation(relation))#返回MyRelation对象存入列表

5）实例化graph连接对象。

数据读取完毕后，我们需要连接neo4j数据库并返回graph对象，然后通过操作graph对象将数据写入数据库。数据库连接参数如下：

• host="localhost"：neo4j服务器的ip地址，本案例中设为本地localhost

• http\_port=7474：服务器端口

• user="neo4j"：数据库user name，如果没有更改过，应该是neo4j

• password="123456"：数据库密码，注意：这里的密码要设为自己的实际密码。

【代码3】build\_medicalgraph.py

graph =Graph(

host="localhost", # neo4j 搭载服务器的ip地址，ifconfig可获取到

http\_port=7474, # neo4j 服务器监听的端口号

user="neo4j", # 数据库user name，如果没有更改过，应该是neo4j

password="123456")

6）存储数据

实例化graph连接对象后，我们既可以将数据写入neo4j数据库了。我们分为两个步骤写入数据，先写入Node，然后再写入Relationship。流程如下：

• 遍历字典node\_dict，调用MyNode对象的get\_node方法实例化Node对象，然后使用graph.create方法写入数据库

• 遍历relationships，实例化startNode和endNode，然后实例化Relationship对象，使用graph.create方法写入数据库

【代码2】build\_medicalgraph.py

#存储节点

for node\_id, node in node\_dict.items():#遍历每一个节点

graph.create(node.get\_node())#存储节点

print("存储节点:",node)

#存储关系

for relationship in relationships:

node1 =node\_dict[relationship.startNodeId].get\_node()#从字典中提取起始节点对象

node2 =node\_dict[relationship.endNodeId].get\_node()#字典中提取终止节点对象

#实例化关系（起始节点，关系类型，终止节点，关系属性）

r = Relationship(node1, relationship.type, node2, \*\*relationship.properties)

#存储关系

graph.create(r)

print("存储关系:",relationship)

8）运行文件，写入数据

运行文件data\_import.py，程序执行，将数据写入数据库，输出结果如下图所示：



7.3 rasa智能对话机器人搭建

本节中，我们需要搭建rasa智能对话机器人的各个部分，这里我们需要说明一下几个术语：

• intents：意图，指的是用户的目的，最终通过文本分类实现

• pipeline：管道，指的是整个对话的流程

• story：Core model 以训练“stories”的形式从真实的会话数据中学习。故事是用户和助手之间的真实对话.

• domain：定义了助手所处的universe：它应该获得的用户输入，应该能够预测的操作，如何响应以及要存储的信息

• actions：机器人助手能够做出的反馈行为

• templates：机器人助手反馈的模板



本节中，我们分别需要编辑以下文件：

• domain.yml：域文件，我们将在该文件中定义整个对话系统需要使用的所有的intents(意图)、entities(实体)、forms(表单)、actions(行为)、responses(响应模板)和slots(槽位)

• nlu.md：语言理解训练数据，我们将在该文件中编写很多用户可能说的话，并标记出其意图以及其中的实体

• stories.md：训练数据集合，原始对话在domain中的映射。Story样本数据就是Rasa Core对话系统要训练的样本，它描述了人机对话过程中可能出现的故事情节，通过对Stories样本和domain的训练得到人机对话系统所需的对话模型。

• actions.py：编写action。当Rasa NLU识别到用户输入信息的意图后，Rasa Core对话管理模块会对其作出回应，回应的操作就是action。

• user\_dict.txt：结巴分词的自定义词典，为了更准确的分词

• config.yml：配置文件，用来配置整个对话系统的pipeline

domain.yml定义了对话机器人应知道的所有信息，相当于大脑框架，指定了意图intents， 实体entities， 插槽slots以及动作actions。

其中，intents和entities与Rasa NLU模型训练样本中标记的一致。slot与标记的entities一致，actions为对话机器人对应用户的请求作出的动作。

此外，domain.yml中的responses部分针对utter\_类型action定义了模板消息，便于对话机器人对相关动作自动回复。

1）查看文件原始内容。

双击domain.yml，可以看到文件中已经有一些默认生成的内容，文件内容如下图所示：



    下面我们来定义系统需要的intents(意图)、entities(实体)、forms(表单)、actions(行为)、responses(响应模板)和slots(槽位)

2）定义intents(意图)。

本系统中，我们需要判断的用户意图有：

inform：提供信息

whattodo：询问能做什么

request\_info：查询信息

greet：打招呼

affirm：确认

thanks：说感谢

restart：重启插叙

request\_case：查询案子

whoareyou：询问身份

goodbye：说再见

deny：拒绝

chitchat：闲聊

stop：

• 在domain.yml文件中编写如下内容：注意，“-“后面有一个空格

【代码25】question\_classifier.py

intents:

- inform

- whattodo

- request\_info

- greet

- affirm

- thanks

- restart

- request\_case

- whoareyou

- goodbye

- deny

- chitchat

- stop

3) 定义entities(实体)

本系统中我们需要设计并提取三种实体：

case：案件，就是案件号

defendant：被告人，就是被告人的姓名

item：属性，职业，住址等

【代码26】question\_classifier.py

entities:

- case

- defendant

- item

4) 定义forms(表单)

本系统中我们需要定义两种表单，不同表单有不同的槽位，执行不同的业务：

case\_form：负责查询案件信息

item\_form：负责查询被告人信息

【代码26】question\_classifier.py

forms:

- case\_form

- item\_form

5) 定义actions(行为)

当Rasa NLU识别到用户输入信息的意图后，Rasa Core对话管理模块会对其作出回应，回应的操作就是action。

Rasa Core支持三种action：

• default actions：默认的一组动作，无需定义，可以直接使用

• action\_listen：监听action

• action\_restart：重置状态

• action\_default\_fallback：当Rasa Core得到的置信度低于设置的阈值时，默认执行该动作。

• utter actions：以utter\_为开头，只发送一条信息给用户作为反馈的动作。定义很简单，只需在domain.yml文件中的actions:字段定义以utter\_为开头即可。具体的回复内容将被定义在templates部分。如果没有utter\_这个前缀，那么action就会被识别为custom actions。

• custom actions：自定义动作，允许开发者执行任何操作并反馈给用户，是action多轮的关键点。需要在domain.yml文件中的actions部分先定义，然后在指定的webserver中实现它。其中，webserver的url地址在endpoint.yml文件中指定。

本系统中，我们定义了以下action：

utter\_answer\_affirm：回复用户的确认

utter\_answer\_deny：回复用户的拒绝

utter\_answer\_goodbye：回复用户的再见

utter\_answer\_greet：回复用户的打招呼

utter\_answer\_restart：回复用户的重启查询

utter\_answer\_thanks：回复用户的感谢

utter\_answer\_whattodo：回复用户询问功能

utter\_answer\_whoareyou：回复用户询问身份

utter\_ask\_case：询问用户查什么案子

utter\_ask\_continue：询问用户是否继续查询

utter\_ask\_item：询问用户差什么信息

utter\_ask\_name：询问用户查谁的信息

utter\_chitchat：回复用户的闲聊

utter\_default：不理解用户意图是的默认回复

utter\_noworries：回复用户的客气

utter\_wrong\_item：无法查询的时候如何回复

【代码26】question\_classifier.py

actions:

- utter\_answer\_affirm

- utter\_answer\_deny

- utter\_answer\_goodbye

- utter\_answer\_greet

- utter\_answer\_restart

- utter\_answer\_thanks

- utter\_answer\_whattodo

- utter\_answer\_whoareyou

- utter\_ask\_case

- utter\_ask\_continue

- utter\_ask\_item

- utter\_ask\_name

- utter\_chitchat

- utter\_default

- utter\_noworries

- utter\_wrong\_item

6) 定义slots(槽位)

插槽是机器人的记忆。它们充当键值存储器，其可用于存储用户提供的信息（例如，他们的家乡）以及关于外部世界收集的信息（例如，数据库查询的结果）。

以查询案件为例，对话机器人必须知道案件号才可以查询，因此在domain.yml中需要在slots部分定义一个插槽，即case。

本系统的功能主要有两个：

查询案件信息：插槽有case

查询被告人信息：插槽有defendant和item

此外我们还需要定义一个槽位requested\_slot作为未特征化的槽位。

【代码26】question\_classifier.py

slots:

case:

auto\_fill: false

type: unfeaturized

defendant:

auto\_fill: false

type: unfeaturized

item:

auto\_fill: false

type: unfeaturized

requested\_slot:

type: unfeaturized

7) 定义responses(响应模板)

responses定义了utter actions具体的回复内容，且每个utter actions下可以定义多条回复信息。当用户发起一个意图，比如"你好！"，就触发utter\_greet操作，Rasa Core会从该action的模板中自动选择其中的一条信息作为结果反馈给用户。

【代码26】question\_classifier.py

responses:

utter\_answer\_affirm:

- text: 嗯嗯，好的！

- text: 嗯嗯，很开心能够帮您解决问题~

- text: 嗯嗯，还需要什么我能够帮助您的呢？

utter\_answer\_deny:

- text: 主人，您不开心吗？不要离开我哦

- text: 怎么了，主人？

utter\_answer\_goodbye:

- text: 再见

- text: 拜拜

- text: 虽然我有万般舍不得，但是天下没有不散的宴席~祝您安好！

- text: 期待下次再见！

- text: 嗯嗯，下次需要时随时记得我哟~

- text: see you!

utter\_answer\_greet:

- text: 您好！请问我可以帮到您吗？

- text: 您好！很高兴为您服务。请说出您要查询的功能？

utter\_answer\_restart:

- text: 好的,已经重置

- text: 好的,已为您重启查询

utter\_answer\_thanks:

- text: 嗯呢。不用客气~

- text: 这是我应该做的，主人~

- text: 嗯嗯，合作愉快！

utter\_answer\_whattodo:

- text: 您好！很高兴为您服务，我目前只支持查询案件信息和被告人信息哦。

utter\_answer\_whoareyou:

- text: 您好！我是小智呀，您的AI智能助理

utter\_ask\_case:

- text: 请问是哪个案件呢?

- text: 哪个案件?

utter\_ask\_continue:

- text: 请问您还要继续吗?

utter\_ask\_item:

- text: 查什么呢?

- text: 你想看什么呢?

- text: 你想看下什么信息呢?

- text: 你想知道些什么?

utter\_ask\_name:

- text: 请问看谁的?

- text: 看下谁的呢?

utter\_chitchat:

- text: 呃呃呃呃呃

- text: 您这是在尬聊吗？

utter\_default:

- text: 没听懂，请换种说法吧~

utter\_noworries:

- text: 不用客气 :)

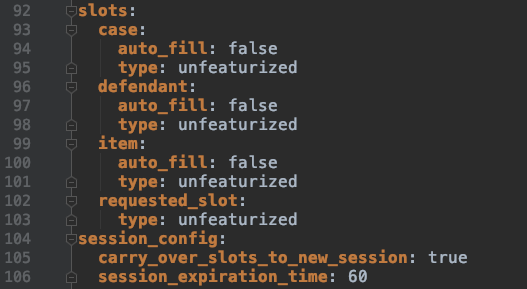
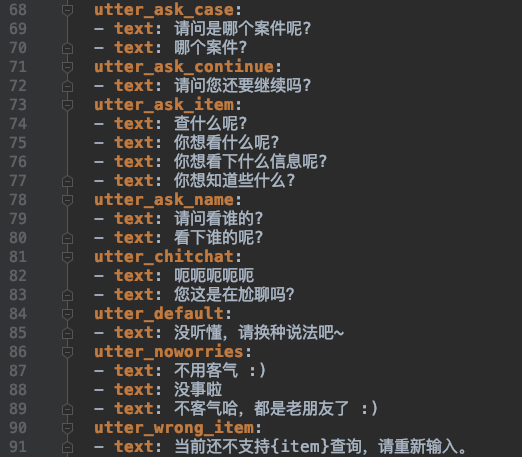
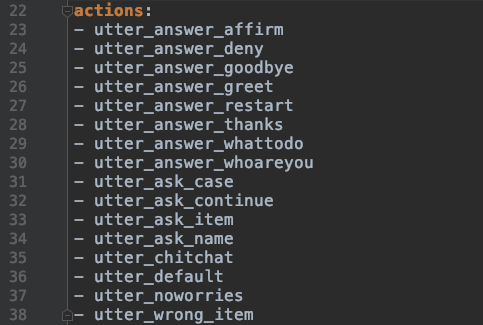
- text: 没事啦

- text: 不客气哈，都是老朋友了 :)

utter\_wrong\_item:

- text: 当前还不支持{item}查询，请重新输入。

至此，domain.yml文件编辑完毕，最终文件内容如下图所示：

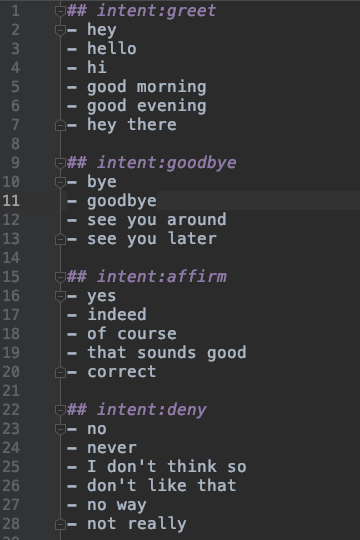


Rasa 助手的第一部分是 NLU 模型，NLU 是自然语言理解，是用来将用户的消息转换为结构化数据。为了让 Rasa 能够做到这一点，你需要提供训练数据，这些训练数据显示 Rasa 助手如何理解用户信息，然后通过这些数据来训练模型。

以 ## 开头的行用来定义你的意图 intents，它们是具有相同含义的消息组。当你向助手发送消息时，Rasa 将预测正确的意图。

1）查看nlu.md文件。

双击nlu.md文件，可以看到文件中已经有一些默认生成的内容，文件内容如下图所示：



文件中已经有一些自动生成的内容，删除掉这些内容，我们需要在该文件中编写训练数据，包括：意图（intent）、实体（entity）和同义词（synonym）

2）定义意图(intent)

意图的作用是来判断用户的目的，是一个文本分类问题，每一个意图就是一个类别，所以我们需要为每个意图提供一些文本数据以便进行机器学习模型的训练。

​    本次案例共提供了以下几种意图分类数据：

greet：表示用户的意图是打招呼

【代码25】question\_parser.py

## intent:greet

- 你好

- 你好啊

- 早上好

- 晚上好

- hello

- hi

- 嗨

- 嗨喽

- 见到你很高兴

- 嘿

- 早

- 上午好

- hello哈喽

- 哈喽哈喽

- hello hello

- 喂喂

- 在吗

goodbye：表示用户的意图是说再见

【代码25】question\_parser.py

## intent:goodbye

- goodbye

- bye

- bye bye

- 88

- 886

- 再见

- 拜

- 拜拜

- 拜拜，下次再聊

- 下次见

- 回头见

- 下次再见

- 下次再聊

- 有空再聊

- 先这样吧

- 好了，就说这么多了

- 好了，先这样

- 没事

whoareou：表示用户的意图是问"你是谁"

【代码25】question\_parser.py

## intent:whoareyou

- 你是谁

- 我知道你吗

- 谁

- 我认识你吗

- 这是谁啊

- 是谁

- 请问你是谁

- 请问我认识你吗

- 你是哪位

- 你是？

- 是谁？

- 可以告诉我你的名字吗

- 你叫什么名字

- 你叫什么名字呢

- 你叫啥名啊?

whattodo：表示用户的意图是想问"你有什么功能"

【代码25】question\_parser.py

## intent:whattodo

- 你支持什么功能

- 你有什么功能

- 你能干什么

- 你能做什么

- 你可以做什么

- 你能干啥

- 你都会啥

- 你会干些啥啊

- 你会做什么

- 你有啥功能

- 你都有啥功能

thanks：表示用户的意思是"谢谢"

【代码25】question\_parser.py

## intent:thanks

- 谢谢

- thanks

- thank you

- 真的太感谢你了，帮了我大忙

- 谢谢你帮了我大忙

- 你帮了我大忙，谢谢你小智

- 非常感谢

- 谢了

- 好的 谢谢

- 好的,谢谢

- 好的,非常感谢

- 好的谢谢

- 不用了, 谢谢

deny：表示用户是想否定和拒绝

【代码25】question\_parser.py

## intent:deny

- 不了

- 算了

- 不用了

- 不需要

- 就这样吧

- 不

- no

- 不可以

- 不是的

- 不认同

- 否定

- 不是这样子的

- 我不同意你的观点

- 不同意

- 不好

- 你长得很美，就不要想得太美。

- 拒绝

- 不行

affirm：表示用户的意思是"确定”

【代码25】question\_parser.py

## intent:affirm

- 知道了

- 是的

- 当然

- 好的

- ok

- 嗯

- 可以

- 你可以这么做

- 你做得可以啊

- 同意

- 听起来不错

- 是这样的

- 的确是这样子的

- 我同意你的观点

- 对的

- 好滴

- 行

- 还行

- 当然可以

- 很好

- 哦哦

chitchat：表示用户在闲聊

【代码25】question\_parser.py

## intent:chitchat

- 你是不是傻

- 你喜欢靓仔靓女吗

- 能跟我说说你的男朋友吗

- 嘿，朋友，你好啊?

- 你是堆的

- 你今天过得开心吗

- 你没事吧

- 今天感觉怎么样

- 广州今天下雨了

- 我能够知道谁邀请你吗

- 请告诉我谁创造了你

- 你的老板是谁

- 我想知道谁发明了你

- 你是哪家公司的

- 你是谁发明的

- 你是什么

- 你还好吗

- 你现在在干什么

- 最近怎么样

- 你最近过得怎么样

- 谁是林俊杰

- 卢十瓦是不是疯了

- 你这个早上干什么了

- 外面真安静

- 现在的生活变得越来越好了

- 是不是下雨了

- 今天的天气很好，不是吗

- 外面好冷啊

- 你认识张天爱吗

- 现在的世界真乱，你说是不是

- 你今天看不起不错哟

- 你觉得华为手机怎么样

- 你知不知道新型冠形病毒

stop：表示用户是在终止对话

**【代码25】question\_parser.py**

## intent:stop

- 你是不是傻

- 你就像个傻逼一样

- 真差劲，什么都不懂

- 真蠢

- 什么都查不到

- 你什么都不会

- 感觉你什么都不知道

- 你太蠢了

- 我不查了，太傻了

- 就这样吧

- 感觉你什么都不会

- 没有了

- 什么都没查到

- 我猜你是不是坏了

request\_case：表示用户想查询案件

【代码25】question\_parser.py

## intent:request\_case

- 查看案件

- 查个案子

- 看个案子

- 看个案件

- 查案子

- 那查个案子吧

- 那帮我查个案子

- 查案件

- 查看一个案子

- 查看一个案件

- 查个case

- 查case

- 查看case

- 查看一个case

- 看一下这个案件的详细信息

- 看下当前案件信息

- [浙1125刑初242号](case)案件

- [浙1125刑初242号](case)

- 查下[浙1125刑初242号](case)案件

- 看下[浙1125刑初242号](case)案件

- 了解一下[浙1125刑初242号](case)案件

- 说下[浙1125刑初242号](case)案件

- 知道[浙1134刑初252号](case)案件吗

- 麻烦看下[浙1125刑初242号](case)案件

- [浙1125刑初242号](case)案件有多少人涉案

- [浙1125刑初242号](case)案件的涉案人数是多少

- 有哪些人参与[浙1125刑初242号](case)案件

- 有多少人涉及[浙1125刑初242号](case)案件

- [浙1125刑初242号](case)案件的涉案人员有哪些

- 查个case把

- [浙1125刑初148号](case)这个案子

- 帮我查个案子

request\_info：表示用户是想查询被告人信息

【代码25】question\_parser.py

## intent:request\_info

- 看个被告人的[信息]{"entity": "item", "value": "\u4e2a\u4eba\u8d44\u6599"}

- 看个人的[信息]{"entity": "item", "value": "\u4e2a\u4eba\u8d44\u6599"}

- 查下被告人[信息]{"entity": "item", "value": "\u4e2a\u4eba\u8d44\u6599"}

- 查下一个人的[信息]{"entity": "item", "value": "\u4e2a\u4eba\u8d44\u6599"}

- 查个人

- 查一个人

- 差一个人

- 叉一个人

- [张远清](defendant)的[出生地](item)是哪里

- [张远清](defendant)在[哪里出生](item)

- [张远清](defendant)的[出生地](item)是哪

- [智航](defendant)的[生日](item)是什么时候

- [张青红](defendant)的[出生日期](item)什么时候

- [智航](defendant)的[出生年月](item)

- [智航](defendant)[何时出生](item)

- [张远清](defendant)[什么时候出生](item)

- [张远清](defendant)[啥时候生的](item)

- 看一下[张远清](defendant)的[个人资料](item)

- 看一下[张远清](defendant)的[资料]{"entity": "item", "value": "\u4e2a\u4eba\u8d44\u6599"}

- 看一下[张远清](defendant)的[情况]{"entity": "item", "value": "\u4e2a\u4eba\u8d44\u6599"}

- 看一下[张远清](defendant)的[数据]{"entity": "item", "value": "\u4e2a\u4eba\u8d44\u6599"}

- 看一下[张远清](defendant)的[信息]{"entity": "item", "value": "\u4e2a\u4eba\u8d44\u6599"}

- [张远清](defendant)是[男还是女](item)

- [张远清](defendant)的[性别](item)

- [张远清](defendant)是[男的](item)吗

- [张远清](defendant)是[女的](item)吗

- [张远清](defendant)是[男的还是女的](item)

- [张远清](defendant)是[女还是男](item)

- [张远清](defendant)是[女的还是男的](item)

- [张远清](defendant)的[户籍所在地](item)是哪

- [张远清](defendant)的[文化程度](item)怎样

- [张远清](defendant)的[文化水平](item)怎样

- [张远清](defendant)[贩毒多少](item)克

- [张远清](defendant)[贩了多少毒](item)

- [张远清](defendant)[卖毒多少](item)

- [张远清](defendant)[卖了多少毒](item)

- [张远清](defendant)的[贩毒量](item)

- [张远清](defendant)的[卖毒量](item)

- [张远清](defendant)是哪个[民族](item)

- [张远清](defendant)是什么[族](item)

- [张远清](defendant)是什么[民族](item)

- [张远清](defendant)是哪个[族](item)

- [张远清](defendant)是那个[族](item)

- [张远清](defendant)是啥[族](item)

- [张远清](defendant)是啥[民族](item)

- [张远清](defendant)的[现住址](item)是哪里

- [张远清](defendant)[住在](item)哪

- [张远清](defendant)[住在](item)什么地方

- [张远清](defendant)的[职业](item)是什么

- [张远清](defendant)是[干啥](item)

- [张远清](defendant)是干什么[职业](item)的

- [张远清](defendant)是[干啥](item)的

- 差个人

- 好吧,帮我查个人

- 查下[陈远清](defendant)的[刑期](item)

- 我想查个被告人[信息]{"entity": "item", "value": "\u4e2a\u4eba\u8d44\u6599"}

- 我想查下[陈远清](defendant)的[信息]{"entity": "item", "value": "\u4e2a\u4eba\u8d44\u6599"}

- 查下[小红](defendant)的[信息]{"entity": "item", "value": "\u4e2a\u4eba\u8d44\u6599"}

- 查下[陈远清](defendant)的[信息]{"entity": "item", "value": "\u4e2a\u4eba\u8d44\u6599"}

- 我还想看下[张青红](defendant)的[情况]{"entity": "item", "value": "\u4e2a\u4eba\u8d44\u6599"}

- 看看[小黑](defendant)的[情况]{"entity": "item", "value": "\u4e2a\u4eba\u8d44\u6599"}

- 帮我查个人

- 查个人的[性别](item)

- 查个人的[职业](item)

- 查个人的[民族](item)

- 查个人的[个人资料](item)

- 看个人

- 看下[陈远清](defendant)[住在](item)什么地方

inform：表示用户在提供信息

【代码25】question\_parser.py

## intent:inform

- [浙1125刑初242号](case)

- [浙1125刑初242号](case)案件

- [职业](item)

- [工作](item)

- [张远清](defendant)

- 什么[文化程度](item)

- [刑期](item)有多长

- 查下[现住址](item)

- 看下[个人资料](item)

- [判了](item)多久

- [判了](item)多长时间

- [智航](defendant)

- 她什么[职业](item)

- [陈远清](defendant)的

- [出生地](item)是哪里

- [哪里出生](item)的

- 她的[出生地](item)是哪

- 他的[生日](item)什么时候

- [何时出生](item)

- [啥时候生的](item)

- [出生日期](item)什么时候

- [出生年月](item)

- [什么时候出生](item)

- [张青红](defendant)

- 他的[个人资料](item)

- 看一下她的[个人资料](item)

- 看一下她的[资料]{"entity": "item", "value": "\u4e2a\u4eba\u8d44\u6599"}

- 看一下她的[情况]{"entity": "item", "value": "\u4e2a\u4eba\u8d44\u6599"}

- 看一下她的[数据]{"entity": "item", "value": "\u4e2a\u4eba\u8d44\u6599"}

- 看一下她的[信息]{"entity": "item", "value": "\u4e2a\u4eba\u8d44\u6599"}

- 看下[张青红](defendant)的

- 看下[智航](defendant)的

- 她是[男还是女](item)

- 他是[男的](item)吗

- 是[女的](item)吗

- [性别](item)

- 他是[男的还是女的](item)

- 他是[女还是男](item)

- 他是[女的还是男的](item)

- 他的[户籍所在地](item)在哪

- 她的[户籍所在地](item)在哪

- [户籍所在地](item)是哪

- [文化程度](item)

- [文化水平](item)

- [文化程度](item)怎样

- [文化水平](item)怎样

- 什么[文化](item)

- 什么[文化水平](item)

- 啥[文化](item)

- 查下[文化水平](item)

- 他[贩毒多少克](item)

- [贩了多少毒](item)

- 他[卖毒多少](item)

- [卖了多少毒](item)

- 他的[贩毒量](item)

- 他的[卖毒量](item)

- [贩毒多少](item)

- 查下[贩了多少毒](item)

- 他是哪个[民族](item)

- 他是什么[族](item)

- 他是什么[民族](item)

- 他是哪个[族](item)

- 他是那个[族](item)

- 他是啥[族](item)

- 他是啥[民族](item)

- 啥[民族](item)

- 什么[民族](item)

- 查下[民族](item)

- [民族](item)

- [名族](item)

- 他的[现住址](item)是哪里

- [现住址](item)是哪里

- 她的[现住址](item)

- 他的[现住址](item)

- 他[住在](item)什么地方

- [住在](item)哪

- [住在](item)什么地方

- 他的[职业](item)是什么

- 是[干啥](item)的

- 他是干什么[职业](item)的

- 干什么[职业](item)

- 干什么[工作](item)

- [干啥](item)的

- 他是[干啥](item)的

- [张清远](defendant)的

- 他的[性别](item)

- [浙1125刑初148号](case)

- 你是[男的还是女的]{"entity": "item", "value": "\u6027\u522b"}

- [陈远清](defendant)

- 他的[职业](item)

- 那查下他的[职业](item)吧

- 再查下[性别](item)

- 他现在[住在]{"entity": "item", "value": "\u73b0\u4f4f\u5740"}哪里呢

- [贩毒量](item)

- [文化程度](item)呢?

- 再看下[名族]{"entity": "item", "value": "\u6c11\u65cf"}

- [小黑](defendant)的

- [职业](item)吧

- 查下[数据](item)

- 那查下[出生地](item)把

- 查下[陈远清](defendant)

- 看下[性别](item)

- 再看下[职业](item)

- 他的[民族](item)呢

- [贩了多少毒]{"entity": "item", "value": "\u8d29\u6bd2\u91cf"}啊他

- 那看下[小明](defendant)的

- 再看下[小红](defendant)的吧

- [小明](defendant)呢

- [小红](defendant)呢

- [川建国](defendant)呢

- 查下[小明](defendant)的把

- 再查下[小黑](defendant)的

- 看下[陈远清](defendant)的

- [明族](item)把

- [哪里生的](item)

- [那里生的](item)

- [住哪儿](item)

- [住哪里](item)

- [住哪](item)呢

- [做啥](item)的

- [干什么](item)的

- 他[住哪里](item)

- 她是[干什么](item)的

- 她是[做什么](item)的

- [住哪](item)的

- 他[住哪](item)呀

- [住哪](item)啊

- 查下[数据]{"entity": "item", "value": "\u4e2a\u4eba\u8d44\u6599"}

- [陈远清](defendant)的

- 随便查下把

- 他[住哪儿]{"entity": "item", "value": "\u73b0\u4f4f\u5740"}的

restart：表示用户想重启查询，系统需要将之前的槽位清空

【代码25】question\_parser.py

## intent:restart

- 重启查询

- 重新查询

- 重置下吧

- 重来

- 再查一次

- 再查一下

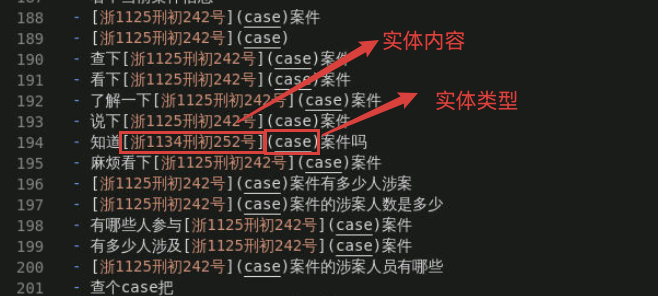
- 重新查

- 那换个人看一下

3）标注实体(entity)

实体就是句子中的关键信息，需要使用命名实体识别算法模型进行提取。本案例提供了了三种类型的实体训练数据：

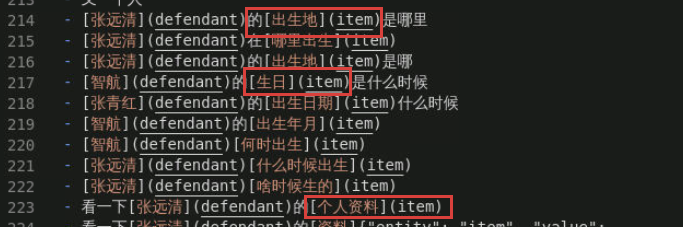
• case：案件号



• defendant：被告人



• item：信息条目



实体已经在意图数据中标注，这里不需要额外编写数据。

4）同义词(synonym)

同义词的主要作用是实现`实体统一`，就是将意思相同的实体词归一为同一个实体词。比如“帮我查一下他的资料”和“帮我查一下他的信息”中的“资料”和“信息”其实是同样的意思，n我们可以利用同义词的形式将他们统一为“个人资料”

【代码26】question\_parser.py

## synonym:个人资料

- 信息

- 资料

- 情况

- 数据

- 个人资料

## synonym:出生地

- 出生地

- 出生在哪里

- 哪里出生

- 出生在什么地方

- 啥地方生的

- 什么地方生的

- 什么地方出生

- 出生在哪

- 哪里生的

- 那里生的

## synonym:刑期

- 刑期

- 判了

## synonym:性别

- 男的还是女的

- 性别

- 男还是女

- 男的

- 女的

- 女还是男

- 女的还是男的

## synonym:文化程度

- 文化程度

- 文化水平

- 文化层次

- 文化

## synonym:民族

- 名族

- 民族

- 明族

## synonym:现住址

- 住在

- 现住址

- 住哪

- 住哪里

- 住哪儿

## synonym:生日

- 生日

- 出生年月

- 出生日期

- 何时出生

- 生的

- 什么时候出生

- 啥时候生的

- 出生在啥时候

## synonym:职业

- 职业

- 工作

- 干啥

- 做啥

- 做什么

- 干什么

## synonym:贩毒量

- 贩了多少毒

- 贩毒量

- 贩毒多少

- 卖毒多少

- 卖毒量

- 卖了多少毒

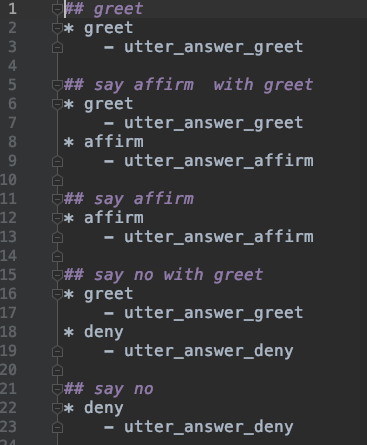
stories可以理解为对话的场景流程，我们需要告诉机器我们的多轮场景是怎样的。Story样本数据就是Rasa Core对话系统要训练的样本，它描述了人机对话过程中可能出现的故事情节，通过对Stories样本和domain的训练得到人机对话系统所需的对话模型。

Stories存储在md文件中。story的符号说明如下：



1）编写类AnswerSearcher及其\_\_init\_\_初始化方法。

由于该文件内容较多，我们再项目准备阶段已经将编写好的文件复制到项目中替换了原来的文件。双击stories.md，，文件内容如下图所示：



注意：这里我们不需要编写文件。

actions.py文件的主要作用是来编写自定义的action，自定义的action可以对查询数据库、调用外部api接口等，通过自定义action可以实现非常强大的功能。

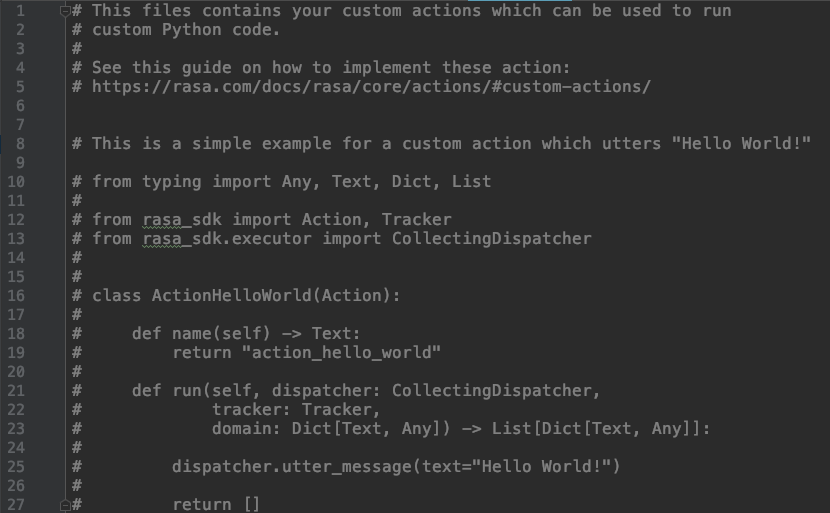
本次案例中我们需要定义两个action，这两个action都继承自FormAction类，实现FormPolicy：

• CaseForm：主要负责案件查询

• ItemForm：主要负责被告人信息查询

1）查看actions.py文件。

双击actions.py文件，可以看到文件中已经有一些默认生成的内容，文件内容如下图所示：



文件中有一些注释内容，是一个默认的action。下面我们来

2）导入需要使用的类

【代码25】question\_parser.py

from typing import Any, Text, Dict, List,Union

from rasa\_sdk import Action, Tracker

from rasa\_sdk.executor import CollectingDispatcher

from rasa\_sdk.forms import FormAction

from rasa\_sdk.events import UserUtteranceReverted, Restarted, SlotSet

from py2neo import Graph, NodeMatcher

3）编写连接neo4j图数据库代码

由于需要查询的数据是存放在neo4j数据库中的，我们需要连接neo4j数据库。

【代码25】question\_parser.py

graph =Graph(

"bolt:0.0.0.0:7687",#连接方式和端口

user="neo4j", # 数据库user name

password="123456")#密码

selector = NodeMatcher(graph)

4）编写CaseForm类

该类继承自FormAction类，负责处理用户查询案件功能，该类有四个方法：

• name方法：返回action的名字。

• required\_slots方法：该方法是静态方法，返回action需要的slots列表，表示该action必须填充的槽位。

• slot\_mappings方法：该方法将槽位映射到实体、意图文本。

• submit方法：该方法将在槽位被填满后调用，在方法中执行action的业务处理代码，一般的操作都在该方法中完成。

首先，编写类的头部和name方法，代码如下：

**【代码25】question\_parser.py**

class CaseForm(FormAction):

def name(self) -> Text:

return "case\_form" #返回form的名字，必须跟domain.yml中定义的一样

编写required\_slots方法，查询case信息，我们需要一个槽位（case：案件名），代码如下：：

**【代码25】question\_parser.py**

@staticmethod

def required\_slots(tracker: Tracker) -> List[Text]:

return ["case"]#案件查询只需要”案件名“一个槽位即可完成业务功能

编写slot\_mappings方法，该方法中我们制定每一个槽位是从哪些意图中提取，提取的实体类型是什么。这里我们指定了是从意图【"inform", "request\_case"】中提取“case”类型实体。代码如下：

【代码25】question\_parser.py

def slot\_mappings(self) -> Dict[Text, Union[Dict, List[Dict]]]:

return {

"case": [

self.from\_entity(

entity="case", intent=["inform", "request\_case"]

),

self.from\_entity(entity="case"),

]

}

编写submit方法，该方法处理流程如下：

• 在tracker中提取槽位case的信息，如果没有则反馈

• 在neo4j数据库中查询对应案件的被告人，返回的是列表

• 判断列表为空，则反馈没有信息

• 如果有，则循环遍历每一个被告人，统计信息

• 调用dispatcher发起utter\_message反馈给用户

代码如下：

【代码25】question\_parser.py

def submit(

self,

dispatcher: CollectingDispatcher,

tracker: Tracker,

domain: Dict[Text, Any],

) -> List[Dict]:

case = xx.get\_slot('case')#从tracker中提取槽值

if not case:#如果槽值为空

dispatcher.utter\_message("服务器开小差了")

return []

all\_defendants = ""

defendants = list(selector.match("被告人", 案件号\_\_contains=case))#查询图数据库

n = defendants.\_\_len\_\_()#计算被告人个数

if n==0:#如果被告人人数为0 表示没有该案件信息

response = "这个系统数据还够完善, 没有找到关于案件'{}'的信息, 抱歉哦..".format(case)

dispatcher.utter\_message(response)#返回指定的文本

return []

for ind in range(n):#对被告人姓名进行拼接

if ind == n - 1:

all\_defendants = all\_defendants + defendants[ind]['name'] + "."

else:

all\_defendants = all\_defendants + defendants[ind]['name'] + ','

response = "{}案件, 有涉案人员:{}, 共{}人".format(case, all\_defendants, n)

dispatcher.utter\_message(response)#返回模板拼接的查询结果

return [SlotSet('case', case)]

5）编写ItemForm类

ItemForm类的编写跟CaseForm类似，ItemForm类处理的是查询被告人信息的业务，该业务中需要的槽位有两个：["defendant","item"]，分别为被告人和属性，即“查询谁的什么信息”。

首先，编写类的头部和name方法，代码如下：

【代码25】question\_parser.py

class ItemForm(FormAction):

def name(self) -> Text:

return "item\_form"

编写required\_slots方法，查询被告人信息，我们需要的槽位有两个：["defendant","item"]，代码如下：

【代码25】question\_parser.py

@staticmethod

def required\_slots(tracker: Tracker) -> List[Text]:

return ["defendant","item"]#查询被告人信息需要[被告人]和[item]两个槽值，即查询被告人的什么

编写slot\_mappings方法，这里我们指定了是从意图【"inform", "request\_case"】中提取“defendant”和“item”类型实体。代码如下：

【代码25】question\_parser.py

def slot\_mappings(self) -> Dict[Text, Union[Dict, List[Dict]]]:

return {

"defendant":[

self.from\_entity(

entity="defendant", intent=["inform", "request\_info"]

),

self.from\_entity(entity="defendant"),

],

"item": [

self.from\_entity(

entity="item", intent=["inform", "request\_info"]

),

self.from\_entity(entity="item"),

]

}

编写submit方法，该方法处理流程如下：

• 在tracker中提取槽位defendant和item的信息，必须两个槽位都有信息才能执行业务

• 在neo4j数据库中查询对应被告人的对应信息并返回结果

• 判断结果为空，则反馈没有信息

• 如果有，则判断查询的是什么信息，根据不同信息使用模块构造不同的回答

• 调用dispatcher发起utter\_message反馈给用户

代码如下：

【代码25】question\_parser.py

def submit(

self,

dispatcher: CollectingDispatcher,

tracker: Tracker,

domain: Dict[Text, Any],

) -> List[Dict]:

defendant = tracker.get\_slot('defendant')#提取defendant槽值

item = tracker.get\_slot('item')#提取item槽值

if (item == None or defendant == None):#有一个槽值为空就不能完成业务

dispatcher.utter\_message("服务器开小差了")

return []

response="这个系统数据还不够完善, 没有找到{}关于'{}'的信息, 抱歉哦..".format(defendant, item)

person = graph.nodes.match("被告人", name=defendant).first()#查询被告人的所有属性

if not person:#如果结果为空，就表示没有这个人的信息

dispatcher.utter\_message(response)

return []

if item.find("个人资料") != -1:#如果是查“个人资料”，就返回所有信息

response = "{},{},{}生,户籍所在:{}, {}程度, 现住{}, 贩毒{}".format(defendant,

person['性别'],

person['出生日期'],

person['户籍所在地'],

person['文化程度'],

person['现住址'],

person['毒品数量'])

elif item.find("出生地") != -1:#查出生地

response = "{}的出生地是{}".format(defendant, person['出生地'])

elif item.find("生日") != -1:

response = "{}的生日是{}".format(defendant, person['出生日期'])

elif item.find("性别") != -1:

response = "{}的性别是:{}".format(defendant, person['性别'])

elif item.find("户籍所在地") != -1:

response = "{}的户籍所在地是:{}".format(defendant, person['户籍所在地'])

elif item.find("文化程度") != -1:

response = "{}的文化程度是:{}".format(defendant, person['文化程度'])

elif item.find("贩毒量") != -1:

response = "{}的贩毒量是:{}".format(defendant, person['毒品数量'])

elif item.find("民族") != -1:

response = "{}的民族是:{}".format(defendant, person['民族'])

elif item.find("现住址") != -1:

response = "{}的现住址是:{}".format(defendant, person['现住址'])

elif item.find("职业") != -1:

response = "{}的职业是:{}".format(defendant, person['职业'])

dispatcher.utter\_message(response)#返回查询结果

return []

本案例中，我们使用的是jieba工具进行中文分词，由于jieba分词在特定业务场景下可能存在偏差，因此需要将业务中使用的特殊词汇定义为词典，提高分词准确率。

在Crime\_QA项目的“data”目录中新建Python File文件“userdict.txt”。并在该文件中输入如下内容：

【代码28】data/userdict.txt

住哪

卖了

贩了

张远清

看一下

查一下

卖毒

看下

查下

了解下

再看下

住在

陈远清

编写完以上的文件后，下面就可以配置Pipeline了。Pipeline是用来指定自然语言理解过程中每一个处理环节使用的工具，具体环节有：

• 分词

• 提取实体

• 意图分类

• 对话管理的policy

每个环节可供选择的工具有很多，所以需要通过配置文件来指定。

1）查看config.yml文件。

双击config.yml文件，可以看到文件中已经有一些默认生成的内容，文件内容如下图所示：



2）写自己的配置

本案例中我们的配置如下：

• 分词：jieba分词+自定义字典

• 实体提取：正则提取和CRF条件随机场提取

• 意图分类：词频特征+tensorflow Embedding文本分类器

• Policy：

• Embedding对话管理策略（TEDPolicy）

• 内存记忆对话管理策略（MemoizationPolicy）

• 表单管理策略（FormPolicy）

• 单一映射对话管理策略（MappingPolicy）

【代码25】config.yml

language: "zh" #语言设置为中文

pipeline:

- name: "JiebaTokenizer" #使用jieba分词器

dictionary\_path: "data" #配置自定义字典进行分词

- name: "RegexFeaturizer" #正则提取器

- name: "CRFEntityExtractor" #CRF条件随机场提取实体

- name: "EntitySynonymMapper" #同义词映射进行实体归一

- name: "CountVectorsFeaturizer" #提取文本词频特征

analyzer: "char\_wb"

min\_ngram: 1

max\_ngram: 4

- name: "EmbeddingIntentClassifier" #tensorflow Embedding文本分类实现意图识别

policies:

- name: TEDPolicy #Embedding对话管理策略

epochs: 100

max\_history: 5

- name: MemoizationPolicy #内存记忆对话管理策略

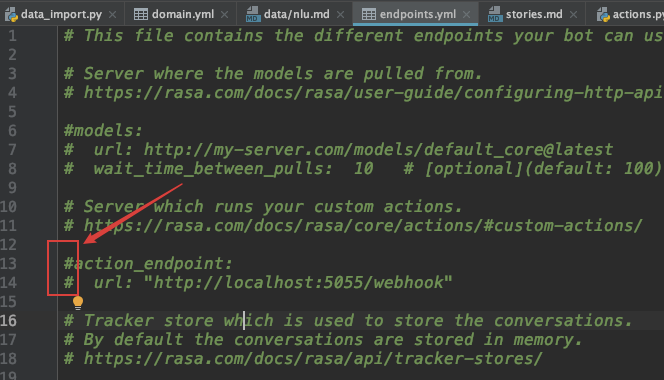
max\_history: 5

- name: FormPolicy #表单管理策略

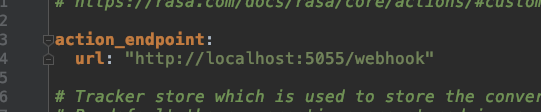
- name: MappingPolicy #单一映射对话管理策略

在后续使用机器人的时候，我们需要启动actions服务，因此我们需要配置服务的地址和端口。这需要在endpoints.yml文件中配置。

双击endpoints.yml文件，打开内容如下，按照提示，去掉action\_endpoint配置项前的"#”：



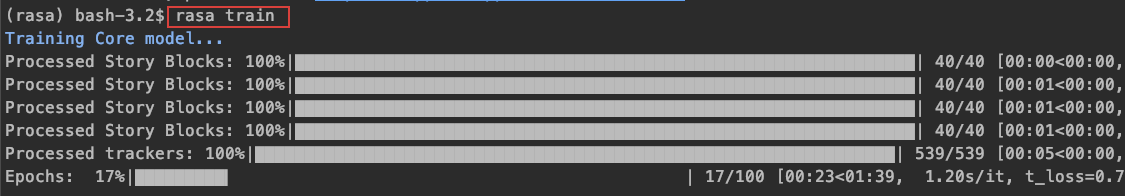
处理后结果如下：



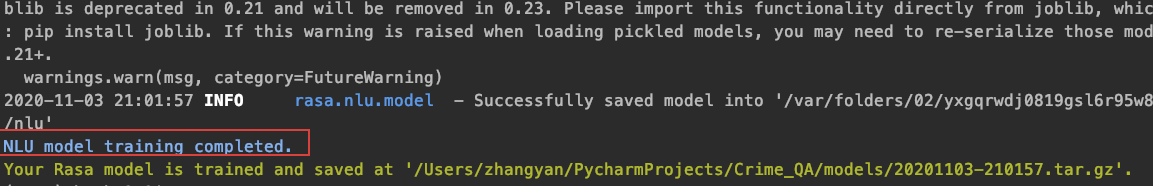
7.4训练和测试机器人

1）训练机器人

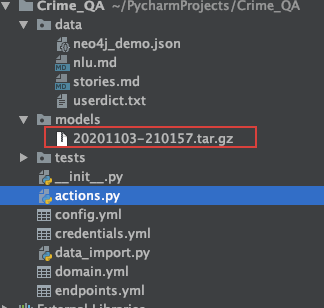
打开pycharm底部的Terminal窗口，输入命令“rasa train”，回车执行，运行结果如下图所示：



稍等一会后，机器人训练结束，结果如下图所示：



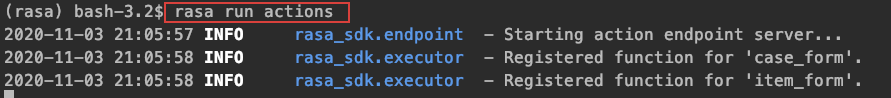
同时，我们可以看到在项目目录下出现一个新的目录“models”，其中存放的是训练好的模型文件，如下图所示：



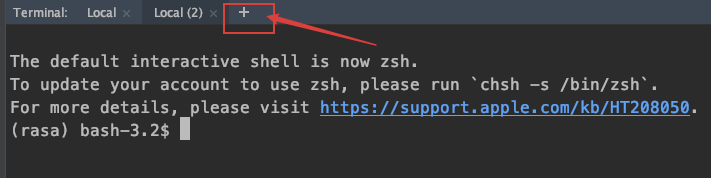
出现该文件说明机器人训练结束。

2）测试机器人

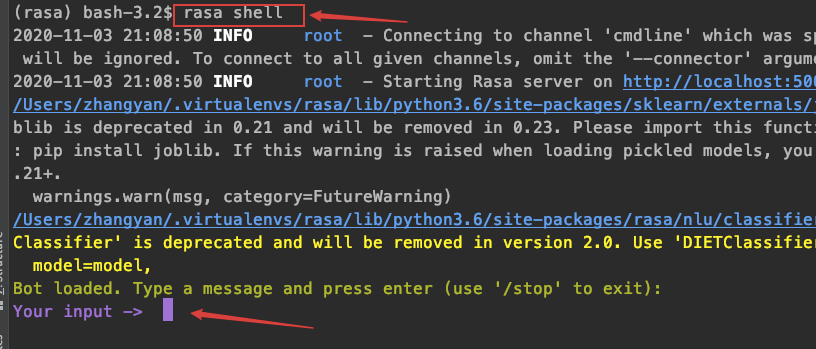
    首先，我们需要启动action，继续在Terminal窗口中输入命令“rasa run actions”并回车执行，启动action服务，结果如下图所示：



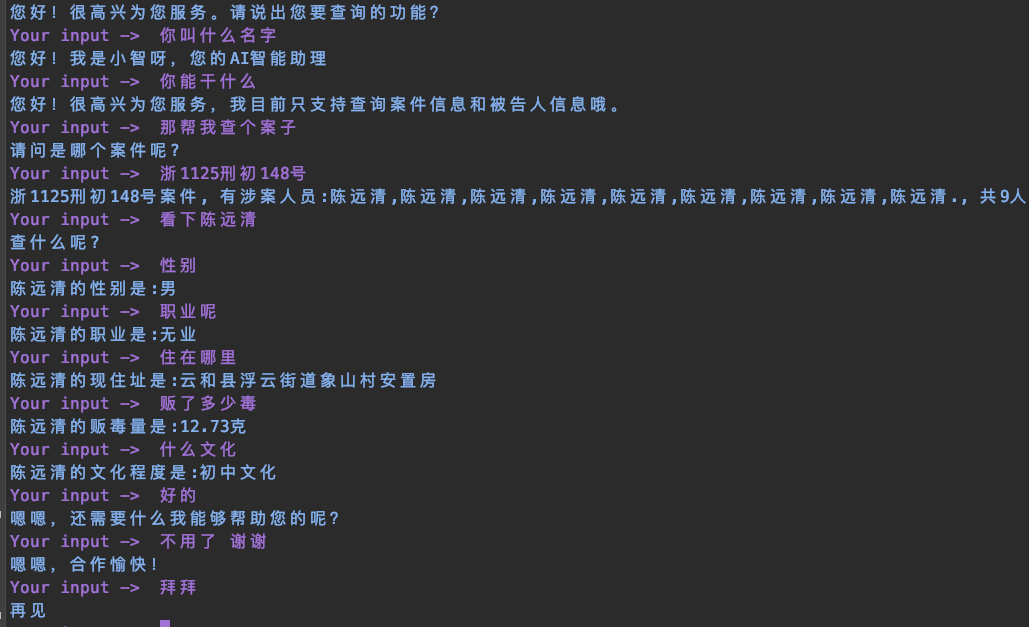
为了简单起见，本案例直接在Terminal中对机器人进行测试，点击Terminal窗口上方的"+"按钮，创建一个新的Terminal窗口：



在新的Terminal窗口中输入命令“rasa shell”，回车执行，稍等程序启动后，出现“Your input—>”提示用户输入，结果如下图所示：



出现上图的结果说明机器人启动成功，我们就可以和机器人进行交流了。我们尝试跟机器人对话，对话过程如下图所示：



8.案例实验结果（结论）

1）知识图谱构建

本案例使用了一份简单的犯罪案件信息数据构建了知识图谱，作为智能对话机器人的知识大脑，为机器人提供知识服务。运行文件data\_import.py，程序执行，将数据写入数据库，输出结果如下图所示：



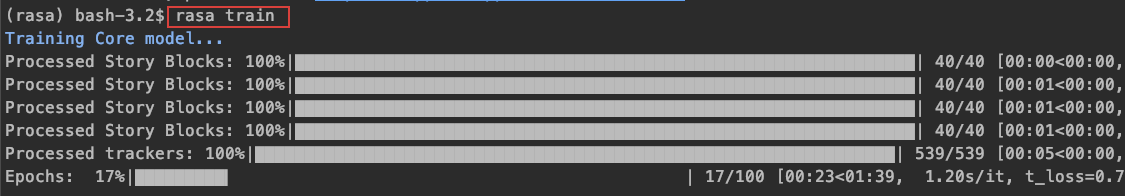
2）训练机器人

我们搭建完整个rasa的各个部分后，就可以使用命令训练机器人，整个过程中我们训练了两个模型：

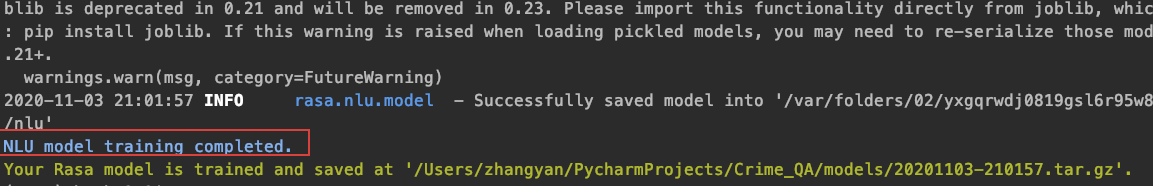
nlu模型：理解用户的意图并提取关键实体信息

DM模型：对话管理模型，用来对多轮对话的上下文进行管理。

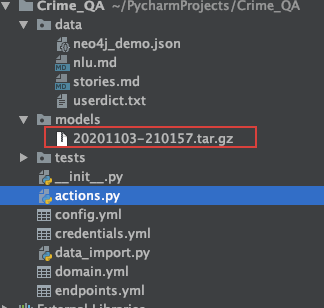
打开pycharm底部的Terminal窗口，输入命令“rasa train”，回车执行，运行结果如下图所示：



稍等一会后，机器人训练结束，结果如下图所示：



同时，我们可以看到在项目目录下出现一个新的目录“models”，其中存放的是训练好的模型文件，如下图所示：

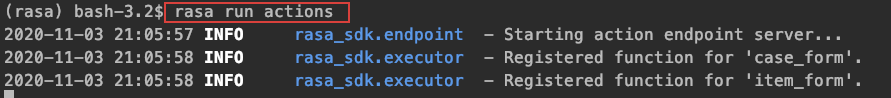


出现该文件说明机器人训练结束。

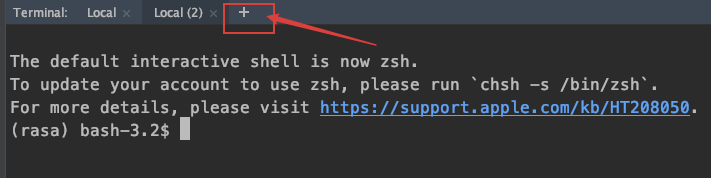
3）测试机器人

机器人训练完成后就可以测试机器人，我们需要分两步进行：启动actions和启动shell。启动actions就是启动了机器人服务，启动shell可以让我们使用terminal来测试机器人。

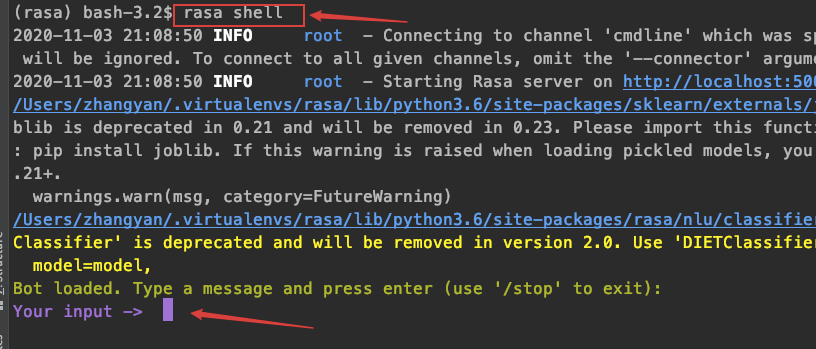
首先，我们需要启动action，继续在Terminal窗口中输入命令“rasa run actions”并回车执行，启动action服务，结果如下图所示：



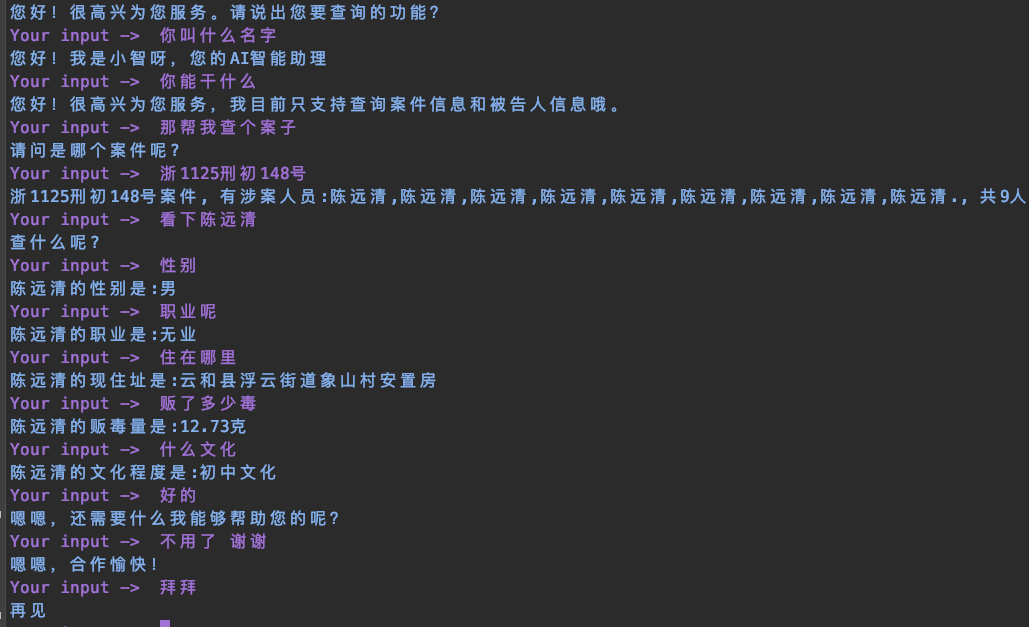
为了简单起见，本案例直接在Terminal中对机器人进行测试，点击Terminal窗口上方的"+"按钮，创建一个新的Terminal窗口：



在新的Terminal窗口中输入命令“rasa shell”，回车执行，稍等程序启动后，出现“Your input—>”提示用户输入，结果如下图所示：



出现上图的结果说明机器人启动成功，我们就可以和机器人进行交流了。我们尝试跟机器人对话，对话过程如下图所示：



9.案例代码

【案例代码】data\_import.py

import json

from py2neo import Graph, Node, Relationship, NodeMatcher

class MyNode(object):

def \_\_init\_\_(self, node\_dict):

self.id = node\_dict.get('id') # 提取id

self.labels = node\_dict.get('labels')#提取标签

self.properties = node\_dict.get('properties')#提取属性

self.properties['id'] = str(self.id)#提取id并存入属性

def \_\_repr\_\_(self) -> str:

return str(self.\_\_dict\_\_)

def get\_node(self):

return Node(\*self.labels, \*\*self.properties)#返回使用数据构造的Node对象

class MyRelation(object):

def \_\_init\_\_(self, relation\_dict):

self.type = relation\_dict.get('type')#提取关系类型

self.startNodeId = str(relation\_dict.get('startNode'))#提取起始节点

self.endNodeId = str(relation\_dict.get('endNode'))#提取终止节点

self.properties = relation\_dict.get('properties')#提取属性

self.properties['id'] = str(relation\_dict.get('id'))#提取id并存入属性

def \_\_repr\_\_(self) -> str:

return str(self.\_\_dict\_\_)

with open('./data/neo4j\_demo.json', 'r', encoding='utf-8') as f:

data = json.loads(f.read())#数据转换成字典

data = data.get('results')[0].get('data')#获取数据

node\_dict = {} # 节点字典

relationships= [] # 关系列表

# 获取所有节点和关系

for \_graph in data:

\_graph = \_graph.get('graph')

for node in \_graph.get('nodes'):#遍历每一个node信息

n = MyNode(node)

node\_dict[str(n.id)] = n#返回节点对象存入字典

for relation in \_graph.get('relationships'):#遍历每一个relationship信息

relationships.append(MyRelation(relation))#返回MyRelation对象存入列表

graph =Graph(

host="localhost", # neo4j 搭载服务器的ip地址，ifconfig可获取到

http\_port=7474, # neo4j 服务器监听的端口号

user="neo4j", # 数据库user name，如果没有更改过，应该是neo4j

password="!@#$1234")

#存储节点

for node\_id, node in node\_dict.items():#遍历每一个节点

graph.create(node.get\_node())#存储节点

print("存储节点:",node)

#存储关系

for relationship in relationships:

node1 =node\_dict[relationship.startNodeId].get\_node()#从字典中提取起始节点对象

node2 =node\_dict[relationship.endNodeId].get\_node()#字典中提取终止节点对象

#实例化关系（起始节点，关系类型，终止节点，关系属性）

r = Relationship(node1, relationship.type, node2, \*\*relationship.properties)

#存储关系

graph.create(r)

print("存储关系:",relationship)

【案例代码】domain.yml

intents:

- inform

- whattodo

- request\_info

- greet

- affirm

- thanks

- restart

- request\_case

- whoareyou

- goodbye

- deny

- chitchat

- stop

entities:

- case

- defendant

- item

forms:

- case\_form

- item\_form

actions:

- utter\_answer\_affirm

- utter\_answer\_deny

- utter\_answer\_goodbye

- utter\_answer\_greet

- utter\_answer\_restart

- utter\_answer\_thanks

- utter\_answer\_whattodo

- utter\_answer\_whoareyou

- utter\_ask\_case

- utter\_ask\_continue

- utter\_ask\_item

- utter\_ask\_name

- utter\_chitchat

- utter\_default

- utter\_noworries

- utter\_wrong\_item

responses:

utter\_answer\_affirm:

- text: 嗯嗯，好的！

- text: 嗯嗯，很开心能够帮您解决问题~

- text: 嗯嗯，还需要什么我能够帮助您的呢？

utter\_answer\_deny:

- text: 主人，您不开心吗？不要离开我哦

- text: 怎么了，主人？

utter\_answer\_goodbye:

- text: 再见

- text: 拜拜

- text: 虽然我有万般舍不得，但是天下没有不散的宴席~祝您安好！

- text: 期待下次再见！

- text: 嗯嗯，下次需要时随时记得我哟~

- text: see you!

utter\_answer\_greet:

- text: 您好！请问我可以帮到您吗？

- text: 您好！很高兴为您服务。请说出您要查询的功能？

utter\_answer\_restart:

- text: 好的,已经重置

- text: 好的,已为您重启查询

utter\_answer\_thanks:

- text: 嗯呢。不用客气~

- text: 这是我应该做的，主人~

- text: 嗯嗯，合作愉快！

utter\_answer\_whattodo:

- text: 您好！很高兴为您服务，我目前只支持查询案件信息和被告人信息哦。

utter\_answer\_whoareyou:

- text: 您好！我是小智呀，您的AI智能助理

utter\_ask\_case:

- text: 请问是哪个案件呢?

- text: 哪个案件?

utter\_ask\_continue:

- text: 请问您还要继续吗?

utter\_ask\_item:

- text: 查什么呢?

- text: 你想看什么呢?

- text: 你想看下什么信息呢?

- text: 你想知道些什么?

utter\_ask\_name:

- text: 请问看谁的?

- text: 看下谁的呢?

utter\_chitchat:

- text: 呃呃呃呃呃

- text: 您这是在尬聊吗？

utter\_default:

- text: 没听懂，请换种说法吧~

utter\_noworries:

- text: 不用客气 :)

- text: 没事啦

- text: 不客气哈，都是老朋友了 :)

utter\_wrong\_item:

- text: 当前还不支持{item}查询，请重新输入。

slots:

case:

auto\_fill: false

type: unfeaturized

defendant:

auto\_fill: false

type: unfeaturized

item:

auto\_fill: false

type: unfeaturized

requested\_slot:

type: unfeaturized

session\_config:

carry\_over\_slots\_to\_new\_session: true

session\_expiration\_time: 60

【案例代码】nlu.md

## intent:greet

- 你好

- 你好啊

- 早上好

- 晚上好

- hello

- hi

- 嗨

- 嗨喽

- 见到你很高兴

- 嘿

- 早

- 上午好

- hello哈喽

- 哈喽哈喽

- hello hello

- 喂喂

- 在吗

## intent:goodbye

- goodbye

- bye

- bye bye

- 88

- 886

- 再见

- 拜

- 拜拜

- 拜拜，下次再聊

- 下次见

- 回头见

- 下次再见

- 下次再聊

- 有空再聊

- 先这样吧

- 好了，就说这么多了

- 好了，先这样

- 没事

## intent:whoareyou

- 你是谁

- 我知道你吗

- 谁

- 我认识你吗

- 这是谁啊

- 是谁

- 请问你是谁

- 请问我认识你吗

- 你是哪位

- 你是？

- 是谁？

- 可以告诉我你的名字吗

- 你叫什么名字

- 你叫什么名字呢

- 你叫啥名啊?

## intent:whattodo

- 你支持什么功能

- 你有什么功能

- 你能干什么

- 你能做什么

- 你可以做什么

- 你能干啥

- 你都会啥

- 你会干些啥啊

- 你会做什么

- 你有啥功能

- 你都有啥功能

## intent:thanks

- 谢谢

- thanks

- thank you

- 真的太感谢你了，帮了我大忙

- 谢谢你帮了我大忙

- 你帮了我大忙，谢谢你小智

- 非常感谢

- 谢了

- 好的 谢谢

- 好的,谢谢

- 好的,非常感谢

- 好的谢谢

- 不用了, 谢谢

## intent:deny

- 不了

- 算了

- 不用了

- 不需要

- 就这样吧

- 不

- no

- 不可以

- 不是的

- 不认同

- 否定

- 不是这样子的

- 我不同意你的观点

- 不同意

- 不好

- 你长得很美，就不要想得太美。

- 拒绝

- 不行

## intent:affirm

- 知道了

- 是的

- 当然

- 好的

- ok

- 嗯

- 可以

- 你可以这么做

- 你做得可以啊

- 同意

- 听起来不错

- 是这样的

- 的确是这样子的

- 我同意你的观点

- 对的

- 好滴

- 行

- 还行

- 当然可以

- 很好

- 哦哦

## intent:chitchat

- 你是不是傻

- 你喜欢靓仔靓女吗

- 能跟我说说你的男朋友吗

- 嘿，朋友，你好啊?

- 你是堆的

- 你今天过得开心吗

- 你没事吧

- 今天感觉怎么样

- 广州今天下雨了

- 我能够知道谁邀请你吗

- 请告诉我谁创造了你

- 你的老板是谁

- 我想知道谁发明了你

- 你是哪家公司的

- 你是谁发明的

- 你是什么

- 你还好吗

- 你现在在干什么

- 最近怎么样

- 你最近过得怎么样

- 谁是林俊杰

- 卢十瓦是不是疯了

- 你这个早上干什么了

- 外面真安静

- 现在的生活变得越来越好了

- 是不是下雨了

- 今天的天气很好，不是吗

- 外面好冷啊

- 你认识张天爱吗

- 现在的世界真乱，你说是不是

- 你今天看不起不错哟

- 你觉得华为手机怎么样

- 你知不知道新型冠形病毒

## intent:stop

- 你是不是傻

- 你就像个傻逼一样

- 真差劲，什么都不懂

- 真蠢

- 什么都查不到

- 你什么都不会

- 感觉你什么都不知道

- 你太蠢了

- 我不查了，太傻了

- 就这样吧

- 感觉你什么都不会

- 没有了

- 什么都没查到

- 我猜你是不是坏了

## intent:request\_case

- 查看案件

- 查个案子

- 看个案子

- 看个案件

- 查案子

- 查案件

- 查看一个案子

- 查看一个案件

- 查个case

- 查case

- 查看case

- 那查个案子吧

- 那帮我查个案子

- 查看一个case

- 看一下这个案件的详细信息

- 看下当前案件信息

- [浙1125刑初242号](case)案件

- [浙1125刑初242号](case)

- 查下[浙1125刑初242号](case)案件

- 看下[浙1125刑初242号](case)案件

- 了解一下[浙1125刑初242号](case)案件

- 说下[浙1125刑初242号](case)案件

- 知道[浙1134刑初252号](case)案件吗

- 麻烦看下[浙1125刑初242号](case)案件

- [浙1125刑初242号](case)案件有多少人涉案

- [浙1125刑初242号](case)案件的涉案人数是多少

- 有哪些人参与[浙1125刑初242号](case)案件

- 有多少人涉及[浙1125刑初242号](case)案件

- [浙1125刑初242号](case)案件的涉案人员有哪些

- 查个case把

- [浙1125刑初148号](case)这个案子

- 帮我查个案子

## intent:request\_info

- 看个被告人的[信息]{"entity": "item", "value": "\u4e2a\u4eba\u8d44\u6599"}

- 看个人的[信息]{"entity": "item", "value": "\u4e2a\u4eba\u8d44\u6599"}

- 查下被告人[信息]{"entity": "item", "value": "\u4e2a\u4eba\u8d44\u6599"}

- 查下一个人的[信息]{"entity": "item", "value": "\u4e2a\u4eba\u8d44\u6599"}

- 查个人

- 查一个人

- 差一个人

- 叉一个人

- [张远清](defendant)的[出生地](item)是哪里

- [张远清](defendant)在[哪里出生](item)

- [张远清](defendant)的[出生地](item)是哪

- [智航](defendant)的[生日](item)是什么时候

- [张青红](defendant)的[出生日期](item)什么时候

- [智航](defendant)的[出生年月](item)

- [智航](defendant)[何时出生](item)

- [张远清](defendant)[什么时候出生](item)

- [张远清](defendant)[啥时候生的](item)

- 看一下[张远清](defendant)的[个人资料](item)

- 看一下[张远清](defendant)的[资料]{"entity": "item", "value": "\u4e2a\u4eba\u8d44\u6599"}

- 看一下[张远清](defendant)的[情况]{"entity": "item", "value": "\u4e2a\u4eba\u8d44\u6599"}

- 看一下[张远清](defendant)的[数据]{"entity": "item", "value": "\u4e2a\u4eba\u8d44\u6599"}

- 看一下[张远清](defendant)的[信息]{"entity": "item", "value": "\u4e2a\u4eba\u8d44\u6599"}

- [张远清](defendant)是[男还是女](item)

- [张远清](defendant)的[性别](item)

- [张远清](defendant)是[男的](item)吗

- [张远清](defendant)是[女的](item)吗

- [张远清](defendant)是[男的还是女的](item)

- [张远清](defendant)是[女还是男](item)

- [张远清](defendant)是[女的还是男的](item)

- [张远清](defendant)的[户籍所在地](item)是哪

- [张远清](defendant)的[文化程度](item)怎样

- [张远清](defendant)的[文化水平](item)怎样

- [张远清](defendant)[贩毒多少](item)克

- [张远清](defendant)[贩了多少毒](item)

- [张远清](defendant)[卖毒多少](item)

- [张远清](defendant)[卖了多少毒](item)

- [张远清](defendant)的[贩毒量](item)

- [张远清](defendant)的[卖毒量](item)

- [张远清](defendant)是哪个[民族](item)

- [张远清](defendant)是什么[族](item)

- [张远清](defendant)是什么[民族](item)

- [张远清](defendant)是哪个[族](item)

- [张远清](defendant)是那个[族](item)

- [张远清](defendant)是啥[族](item)

- [张远清](defendant)是啥[民族](item)

- [张远清](defendant)的[现住址](item)是哪里

- [张远清](defendant)[住在](item)哪

- [张远清](defendant)[住在](item)什么地方

- [张远清](defendant)的[职业](item)是什么

- [张远清](defendant)是[干啥](item)

- [张远清](defendant)是干什么[职业](item)的

- [张远清](defendant)是[干啥](item)的

- 差个人

- 好吧,帮我查个人

- 查下[陈远清](defendant)的[刑期](item)

- 我想查个被告人[信息]{"entity": "item", "value": "\u4e2a\u4eba\u8d44\u6599"}

- 我想查下[陈远清](defendant)的[信息]{"entity": "item", "value": "\u4e2a\u4eba\u8d44\u6599"}

- 查下[小红](defendant)的[信息]{"entity": "item", "value": "\u4e2a\u4eba\u8d44\u6599"}

- 查下[陈远清](defendant)的[信息]{"entity": "item", "value": "\u4e2a\u4eba\u8d44\u6599"}

- 我还想看下[张青红](defendant)的[情况]{"entity": "item", "value": "\u4e2a\u4eba\u8d44\u6599"}

- 看看[小黑](defendant)的[情况]{"entity": "item", "value": "\u4e2a\u4eba\u8d44\u6599"}

- 帮我查个人

- 查个人的[性别](item)

- 查个人的[职业](item)

- 查个人的[民族](item)

- 查个人的[个人资料](item)

- 看个人

- 看下[陈远清](defendant)[住在](item)什么地方

## intent:inform

- [浙1125刑初242号](case)

- [浙1125刑初242号](case)案件

- [职业](item)

- [工作](item)

- [张远清](defendant)

- 什么[文化程度](item)

- [刑期](item)有多长

- 查下[现住址](item)

- 看下[个人资料](item)

- [判了](item)多久

- [判了](item)多长时间

- [智航](defendant)

- 她什么[职业](item)

- [陈远清](defendant)的

- [出生地](item)是哪里

- [哪里出生](item)的

- 她的[出生地](item)是哪

- 他的[生日](item)什么时候

- [何时出生](item)

- [啥时候生的](item)

- [出生日期](item)什么时候

- [出生年月](item)

- [什么时候出生](item)

- [张青红](defendant)

- 他的[个人资料](item)

- 看一下她的[个人资料](item)

- 看一下她的[资料]{"entity": "item", "value": "\u4e2a\u4eba\u8d44\u6599"}

- 看一下她的[情况]{"entity": "item", "value": "\u4e2a\u4eba\u8d44\u6599"}

- 看一下她的[数据]{"entity": "item", "value": "\u4e2a\u4eba\u8d44\u6599"}

- 看一下她的[信息]{"entity": "item", "value": "\u4e2a\u4eba\u8d44\u6599"}

- 看下[张青红](defendant)的

- 看下[智航](defendant)的

- 她是[男还是女](item)

- 他是[男的](item)吗

- 是[女的](item)吗

- [性别](item)

- 他是[男的还是女的](item)

- 他是[女还是男](item)

- 他是[女的还是男的](item)

- 他的[户籍所在地](item)在哪

- 她的[户籍所在地](item)在哪

- [户籍所在地](item)是哪

- [文化程度](item)

- [文化水平](item)

- [文化程度](item)怎样

- [文化水平](item)怎样

- 什么[文化](item)

- 什么[文化水平](item)

- 啥[文化](item)

- 查下[文化水平](item)

- 他[贩毒多少克](item)

- [贩了多少毒](item)

- 他[卖毒多少](item)

- [卖了多少毒](item)

- 他的[贩毒量](item)

- 他的[卖毒量](item)

- [贩毒多少](item)

- 查下[贩了多少毒](item)

- 他是哪个[民族](item)

- 他是什么[族](item)

- 他是什么[民族](item)

- 他是哪个[族](item)

- 他是那个[族](item)

- 他是啥[族](item)

- 他是啥[民族](item)

- 啥[民族](item)

- 什么[民族](item)

- 查下[民族](item)

- [民族](item)

- [名族](item)

- 他的[现住址](item)是哪里

- [现住址](item)是哪里

- 她的[现住址](item)

- 他的[现住址](item)

- 他[住在](item)什么地方

- [住在](item)哪

- [住在](item)什么地方

- 他的[职业](item)是什么

- 是[干啥](item)的

- 他是干什么[职业](item)的

- 干什么[职业](item)

- 干什么[工作](item)

- [干啥](item)的

- 他是[干啥](item)的

- [张清远](defendant)的

- 他的[性别](item)

- [浙1125刑初148号](case)

- 你是[男的还是女的]{"entity": "item", "value": "\u6027\u522b"}

- [陈远清](defendant)

- 他的[职业](item)

- 那查下他的[职业](item)吧

- 再查下[性别](item)

- 他现在[住在]{"entity": "item", "value": "\u73b0\u4f4f\u5740"}哪里呢

- [贩毒量](item)

- [文化程度](item)呢?

- 再看下[名族]{"entity": "item", "value": "\u6c11\u65cf"}

- [小黑](defendant)的

- [职业](item)吧

- 查下[数据](item)

- 那查下[出生地](item)把

- 查下[陈远清](defendant)

- 看下[性别](item)

- 再看下[职业](item)

- 他的[民族](item)呢

- [贩了多少毒]{"entity": "item", "value": "\u8d29\u6bd2\u91cf"}啊他

- 那看下[小明](defendant)的

- 再看下[小红](defendant)的吧

- [小明](defendant)呢

- [小红](defendant)呢

- [川建国](defendant)呢

- 查下[小明](defendant)的把

- 再查下[小黑](defendant)的

- 看下[陈远清](defendant)的

- [明族](item)把

- [哪里生的](item)

- [那里生的](item)

- [住哪儿](item)

- [住哪里](item)

- [住哪](item)呢

- [做啥](item)的

- [干什么](item)的

- 他[住哪里](item)

- 她是[干什么](item)的

- 她是[做什么](item)的

- [住哪](item)的

- 他[住哪](item)呀

- [住哪](item)啊

- 查下[数据]{"entity": "item", "value": "\u4e2a\u4eba\u8d44\u6599"}

- [陈远清](defendant)的

- 随便查下把

- 他[住哪儿]{"entity": "item", "value": "\u73b0\u4f4f\u5740"}的

## intent:restart

- 重启查询

- 重新查询

- 重置下吧

- 重来

- 再查一次

- 再查一下

- 重新查

- 那换个人看一下

## synonym:个人资料

- 信息

- 资料

- 情况

- 数据

- 个人资料

## synonym:出生地

- 出生地

- 出生在哪里

- 哪里出生

- 出生在什么地方

- 啥地方生的

- 什么地方生的

- 什么地方出生

- 出生在哪

- 哪里生的

- 那里生的

## synonym:刑期

- 刑期

- 判了

## synonym:性别

- 男的还是女的

- 性别

- 男还是女

- 男的

- 女的

- 女还是男

- 女的还是男的

## synonym:文化程度

- 文化程度

- 文化水平

- 文化层次

- 文化

## synonym:民族

- 名族

- 民族

- 明族

## synonym:现住址

- 住在

- 现住址

- 住哪

- 住哪里

- 住哪儿

## synonym:生日

- 生日

- 出生年月

- 出生日期

- 何时出生

- 生的

- 什么时候出生

- 啥时候生的

- 出生在啥时候

## synonym:职业

- 职业

- 工作

- 干啥

- 做啥

- 做什么

- 干什么

## synonym:贩毒量

- 贩了多少毒

- 贩毒量

- 贩毒多少

- 卖毒多少

- 卖毒量

- 卖了多少毒

【案例代码】stories.md

## greet

\* greet

- utter\_answer\_greet

## say affirm with greet

\* greet

- utter\_answer\_greet

\* affirm

- utter\_answer\_affirm

## say affirm

\* affirm

- utter\_answer\_affirm

## say no with greet

\* greet

- utter\_answer\_greet

\* deny

- utter\_answer\_deny

## say no

\* deny

- utter\_answer\_deny

## say goodbye

\* goodbye

- utter\_answer\_goodbye

## thanks with greet

\* greet

- utter\_answer\_greet

\* thanks

- utter\_answer\_thanks

## thanks

\* thanks

- utter\_answer\_thanks

## who are you with greet

\* greet

- utter\_answer\_greet

\* whoareyou

- utter\_answer\_whoareyou

## who are you

\* whoareyou

- utter\_answer\_whoareyou

## who are you with greet

\* greet

- utter\_answer\_greet

\* whoareyou

- utter\_answer\_whoareyou

## what to do

\* whattodo

- utter\_answer\_whattodo

## what to do with greet

\* greet

- utter\_answer\_greet

\* whattodo

- utter\_answer\_whattodo

## interactive\_story\_3

# happy path：item+item+item+item+item

\* request\_info{"defendant": "张青红"}

- item\_form

- form{"name": "item\_form"}

- slot{"defendant": "张青红"}

- slot{"requested\_slot": "item"}

\* form: request\_info{"item": "性别"}

- form: item\_form

- slot{"item": "性别"}

- slot{"defendant": "张青红"}

- form{"name": null}

- slot{"requested\_slot": null}

\* inform{"item": "出生地"}

- item\_form

- form{"name": "item\_form"}

- slot{"defendant": "张青红"}

- slot{"item": "出生地"}

- slot{"item": null}

- form{"name": null}

- slot{"requested\_slot": null}

\* inform{"item": "生日"}

- item\_form

- form{"name": "item\_form"}

- slot{"defendant": "张青红"}

- slot{"item": "生日"}

- slot{"item": null}

- form{"name": null}

- slot{"requested\_slot": null}

\* inform{"item": "文化程度"}

- item\_form

- form{"name": "item\_form"}

- slot{"defendant": "张青红"}

- slot{"item": "文化程度"}

- slot{"item": null}

- form{"name": null}

- slot{"requested\_slot": null}

\* affirm

- utter\_answer\_affirm

## interactive\_story\_4

# happy path：(defendant+item)+item+item

\* request\_info{"item": "性别"}

- item\_form

- form{"name": "item\_form"}

- slot{"item": "性别"}

- slot{"defendant": "张青红"}

- form{"name": null}

- slot{"requested\_slot": null}

\* inform{"item": "出生地"}

- item\_form

- form{"name": "item\_form"}

- slot{"defendant": "张青红"}

- slot{"item": "出生地"}

- slot{"item": null}

- form{"name": null}

- slot{"requested\_slot": null}

\* inform{"item": "生日"}

- item\_form

- form{"name": "item\_form"}

- slot{"defendant": "张青红"}

- slot{"item": "生日"}

- slot{"item": null}

- form{"name": null}

- slot{"requested\_slot": null}

\* inform{"item": "文化程度"}

- item\_form

- form{"name": "item\_form"}

- slot{"defendant": "张青红"}

- slot{"item": "文化程度"}

- slot{"item": null}

- form{"name": null}

- slot{"requested\_slot": null}

\* affirm

- utter\_answer\_affirm

## interactive\_story\_5

# unhappy：chit+answer

\* greet

- utter\_answer\_greet

\* request\_info{"defendant": "张青红"}

- item\_form

- form{"name": "item\_form"}

- slot{"defendant": "张青红"}

- slot{"item": null}

- slot{"requested\_slot": "item"}

\* chitchat

- utter\_chitchat

- item\_form

- slot{"requested\_slot": "item"}

\* form: request\_info{"item": "性别"}

- form: item\_form

- slot{"item": "性别"}

- slot{"defendant": "张青红"}

- form{"name": null}

- slot{"requested\_slot": null}

\* thanks

- utter\_noworries

## interactive\_story\_6

# very unhappy：chit + chit + answer

\* greet

- utter\_answer\_greet

\* request\_info{"defendant": "张清远"}

- item\_form

- form{"name": "item\_form"}

- slot{"defendant": "张清远"}

- slot{"item": null}

- slot{"requested\_slot": "item"}

\* chitchat

- utter\_chitchat

- item\_form

- slot{"requested\_slot": "item"}

\* chitchat

- utter\_chitchat

- item\_form

- slot{"requested\_slot": "item"}

\* form: request\_info{"item": "性别"}

- form: item\_form

- slot{"item": "性别"}

- slot{"defendant": "张清远"}

- form{"name": null}

- slot{"requested\_slot": null}

\* thanks

- utter\_noworries

## interactive\_story\_7

# very very unhappy：chitchat+chitchat+chitchat+answer

\* greet

- utter\_answer\_greet

\* request\_info{"defendant": "张清远"}

- item\_form

- form{"name": "item\_form"}

- slot{"defendant": "张清远"}

- slot{"item": null}

- slot{"requested\_slot": "item"}

\* chitchat

- utter\_chitchat

- item\_form

- slot{"requested\_slot": "item"}

\* chitchat

- utter\_chitchat

- item\_form

- slot{"requested\_slot": "item"}

\* chitchat

- utter\_chitchat

- item\_form

- slot{"requested\_slot": "item"}

\* form: request\_info{"item": "性别"}

- form: item\_form

- slot{"item": "性别"}

- slot{"defendant": "张清远"}

- form{"name": null}

- slot{"requested\_slot": null}

\* thanks

- utter\_noworries

## interactive\_story\_8

## unhappy path：stop and really stop path

\* greet

- utter\_answer\_greet

\* request\_info{"defendant": "张清远"}

- item\_form

- form{"name": "item\_form"}

- slot{"defendant": "张清远"}

- slot{"item": null}

- slot{"requested\_slot": "item"}

\* stop

- utter\_ask\_continue

\* deny

- action\_deactivate\_form

- form{"name": null}

- slot{"requested\_slot": null}

\* thanks

- utter\_noworries

## interactive\_story\_9

## unhappy path：stop but continue path

\* greet

- utter\_answer\_greet

\* request\_info{"defendant": "张清远"}

- item\_form

- form{"name": "item\_form"}

- slot{"defendant": "张清远"}

- slot{"item": null}

- slot{"requested\_slot": "item"}

\* stop

- utter\_ask\_continue

\* affirm

- item\_form

- slot{"requested\_slot": "item"}

\* form: request\_info{"item": "生日"}

- form: item\_form

- slot{"item": "生日"}

- slot{"defendant": "张清远"}

- form{"name": null}

- slot{"requested\_slot": null}

\* thanks

- utter\_noworries

## interactive\_story\_10

# unhappy path：chitchat stop but continue path

\* greet

- utter\_answer\_greet

\* request\_info{"defendant": "张清远"}

- item\_form

- form{"name": "item\_form"}

- slot{"defendant": "张清远"}

- slot{"item": null}

- slot{"requested\_slot": "item"}

\* chitchat

- utter\_chitchat

- item\_form

- slot{"requested\_slot": "item"}

\* stop

- utter\_ask\_continue

\* affirm

- item\_form

- slot{"requested\_slot": "item"}

\* form: request\_info{"item": "职业"}

- form: item\_form

- slot{"item": "职业"}

- slot{"defendant": "张清远"}

- form{"name": null}

- slot{"requested\_slot": null}

\* thanks

- utter\_noworries

## interactive\_story\_11

## unhappy path: stop but continue and chitchat path

\* greet

- utter\_answer\_greet

\* request\_info{"defendant": "张清远"}

- item\_form

- form{"name": "item\_form"}

- slot{"defendant": "张清远"}

- slot{"item": null}

- slot{"requested\_slot": "item"}

\* stop

- utter\_ask\_continue

\* affirm

- item\_form

- slot{"requested\_slot": "item"}

\* chitchat

- utter\_chitchat

- item\_form

- slot{"requested\_slot": "item"}

\* form: request\_info{"item": "民族"}

- form: item\_form

- slot{"item": "民族"}

- slot{"defendant": "张清远"}

- form{"name": null}

- slot{"requested\_slot": null}

\* thanks

- utter\_noworries

## interactive\_story\_12

## unhappy path: chitchat stop but continue and chitchat path

\* greet

- utter\_answer\_greet

\* request\_info{"defendant": "张清远"}

- item\_form

- form{"name": "item\_form"}

- slot{"defendant": "张清远"}

- slot{"item": null}

- slot{"requested\_slot": "item"}

\* chitchat

- utter\_chitchat

- item\_form

- slot{"requested\_slot": "item"}

\* stop

- utter\_ask\_continue

\* affirm

- item\_form

- slot{"requested\_slot": "item"}

\* chitchat

- utter\_chitchat

- item\_form

- slot{"requested\_slot": "item"}

\* form: request\_info{"item": "刑期"}

- form: item\_form

- slot{"item": "刑期"}

- slot{"defendant": "张清远"}

- form{"name": null}

- slot{"requested\_slot": null}

\* thanks

- utter\_noworries

## interactive\_story\_13

## unhappy path: chitchat, stop and really stop path

\* greet

- utter\_answer\_greet

\* request\_info

- utter\_ask\_name

- action\_listen

\* form: inform{"defendant": "张清远"}

- item\_form

- slot{"defendant": "张清远"}

- slot{"requested\_slot": "item"}

\* chitchat

- utter\_chitchat

- item\_form

- slot{"requested\_slot": "item"}

\* stop

- utter\_ask\_continue

\* deny

- action\_deactivate\_form

- form{"name": null}

- slot{"requested\_slot": null}

\* thanks

- utter\_noworries

## interactive\_story\_15

\* greet

- utter\_answer\_greet

\* request\_info

- utter\_ask\_name

- action\_listen

\* form: inform{"defendant": "张清远"}

- item\_form

- slot{"defendant": "张清远"}

- slot{"requested\_slot": "item"}

\* form: inform{"item": "性别"}

- form: item\_form

## interactive\_story\_16

\* greet

- utter\_answer\_greet

\* request\_case

- case\_form

- form{"name": "case\_form"}

- slot{"requested\_slot": "case"}

\* form: inform{"case": "浙1125刑初148号"}

- form: case\_form

- slot{"case": "浙1125刑初148号"}

- form{"name": null}

- slot{"requested\_slot": null}

\* affirm

- utter\_answer\_affirm

\* whoareyou

- utter\_answer\_whoareyou

\* inform{"item": "性别"}

- utter\_chitchat

- item\_form

- form{"name": "item\_form"}

- slot{"requested\_slot": "defendant"}

\* request\_info

- utter\_ask\_name

- action\_listen

\* form: inform{"defendant": "陈远清"}

- item\_form

- slot{"defendant": "陈远清"}

- slot{"requested\_slot": "item"}

\* form: inform{"item": "职业"}

- form: item\_form

- slot{"item": "职业"}

- form{"name": null}

- slot{"requested\_slot": null}

\* thanks

- utter\_answer\_thanks

## interactive\_story\_17

\* greet

- utter\_answer\_greet

\* whoareyou

- utter\_answer\_whoareyou

\* inform

- utter\_answer\_affirm

\* request\_case

- case\_form

- form{"name": "case\_form"}

- slot{"requested\_slot": "case"}

\* form: request\_case{"case": "浙1125刑初148号"}

- form: case\_form

- slot{"case": "浙1125刑初148号"}

- slot{"case": "浙1125刑初148号"}

- form{"name": null}

- slot{"requested\_slot": null}

\* thanks

- utter\_noworries

\* goodbye

- utter\_answer\_goodbye

## interactive\_story\_18

\* greet

- utter\_answer\_greet

\* request\_info{"defendant": "陈远清", "item": "刑期"}

- item\_form

- form{"name": "item\_form"}

- slot{"defendant": "陈远清"}

- slot{"item": "刑期"}

- form{"name": null}

- slot{"requested\_slot": null}

\* inform{"item": "职业"}

- item\_form

- form{"name": "item\_form"}

- slot{"defendant": "陈远清"}

- slot{"item": "刑期"}

- slot{"item": "职业"}

- form{"name": null}

- slot{"requested\_slot": null}

\* inform{"item": "性别"}

- item\_form

- form{"name": "item\_form"}

- slot{"defendant": "陈远清"}

- slot{"item": "职业"}

- slot{"item": "性别"}

- form{"name": null}

- slot{"requested\_slot": null}

\* inform{"item": "现住址"}

- item\_form

- form{"name": "item\_form"}

- slot{"defendant": "陈远清"}

- slot{"item": "性别"}

- slot{"item": "现住址"}

- form{"name": null}

- slot{"requested\_slot": null}

\* inform{"item": "贩毒量"}

- item\_form

- form{"name": "item\_form"}

- slot{"defendant": "陈远清"}

- slot{"item": "现住址"}

- slot{"item": "贩毒量"}

- form{"name": null}

- slot{"requested\_slot": null}

\* inform{"item": "文化程度"}

- item\_form

- form{"name": "item\_form"}

- slot{"defendant": "陈远清"}

- slot{"item": "贩毒量"}

- slot{"item": "文化程度"}

- form{"name": null}

- slot{"requested\_slot": null}

\* inform{"item": "民族"}

- item\_form

- form{"name": "item\_form"}

- slot{"defendant": "陈远清"}

- slot{"item": "文化程度"}

- slot{"item": "民族"}

- form{"name": null}

- slot{"requested\_slot": null}

\* thanks

- utter\_noworries

## interactive\_story\_19

\* greet

- utter\_answer\_greet

\* request\_info

- utter\_ask\_name

\* inform{"defendant": "小黑"}

- item\_form

- form{"name": "item\_form"}

- slot{"defendant": "小黑"}

- slot{"requested\_slot": "item"}

\* form: inform{"item": "职业"}

- form: item\_form

- slot{"item": "职业"}

- form{"name": null}

- slot{"requested\_slot": null}

\* affirm

- utter\_answer\_affirm

## interactive\_story\_20

\* greet

- utter\_answer\_greet

\* request\_info{"defendant": "陈远清", "item": "个人资料"}

- item\_form

- form{"name": "item\_form"}

- slot{"defendant": "陈远清"}

- slot{"item": "个人资料"}

- form{"name": null}

- slot{"requested\_slot": null}

\* inform{"item": "数据"}

- item\_form

- form{"name": "item\_form"}

- slot{"defendant": "陈远清"}

- slot{"item": "个人资料"}

- slot{"item": "个人资料"}

- form{"name": null}

- slot{"requested\_slot": null}

\* inform{"item": "出生地"}

- item\_form

- form{"name": "item\_form"}

- slot{"defendant": "陈远清"}

- slot{"item": "个人资料"}

- slot{"item": "出生地"}

- form{"name": null}

- slot{"requested\_slot": null}

\* affirm

- utter\_answer\_affirm

## interactive\_story\_21

\* greet

- utter\_answer\_greet

\* inform{"defendant": "陈远清"}

- item\_form

- form{"name": "item\_form"}

- slot{"defendant": "陈远清"}

- slot{"requested\_slot": "item"}

\* form: inform{"item": "性别"}

- form: item\_form

- slot{"item": "性别"}

- form{"name": null}

- slot{"requested\_slot": null}

\* inform{"item": "职业"}

- item\_form

- form{"name": "item\_form"}

- slot{"defendant": "陈远清"}

- slot{"item": "性别"}

- slot{"item": "职业"}

- form{"name": null}

- slot{"requested\_slot": null}

\* inform{"item": "民族"}

- item\_form

- form{"name": "item\_form"}

- slot{"defendant": "陈远清"}

- slot{"item": "职业"}

- slot{"item": "民族"}

- form{"name": null}

- slot{"requested\_slot": null}

\* inform{"item": "贩毒量"}

- item\_form

- form{"name": "item\_form"}

- slot{"defendant": "陈远清"}

- slot{"item": "民族"}

- slot{"item": "贩毒量"}

- form{"name": null}

- slot{"requested\_slot": null}

\* affirm

- utter\_answer\_affirm

## interactive\_story\_22

\* greet

- utter\_answer\_greet

\* request\_info{"defendant": "小红", "item": "个人资料"}

- item\_form

- form{"name": "item\_form"}

- slot{"defendant": "小红"}

- slot{"item": "个人资料"}

- form{"name": null}

- slot{"requested\_slot": null}

\* inform{"defendant": "小明"}

- item\_form

- form{"name": "item\_form"}

- slot{"defendant": "小红"}

- slot{"item": "个人资料"}

- slot{"defendant": "小明"}

- form{"name": null}

- slot{"requested\_slot": null}

\* restart

- action\_restart

- utter\_answer\_restart

\* request\_info{"defendant": "陈远清", "item": "个人资料"}

- item\_form

- form{"name": "item\_form"}

- slot{"defendant": "陈远清"}

- slot{"item": "个人资料"}

- form{"name": null}

- slot{"requested\_slot": null}

\* inform{"item": "性别"}

- item\_form

- form{"name": "item\_form"}

- slot{"defendant": "陈远清"}

- slot{"item": "个人资料"}

- slot{"item": "性别"}

- form{"name": null}

- slot{"requested\_slot": null}

\* restart

- action\_restart

- utter\_answer\_restart

\* request\_info{"defendant": "张青红", "item": "个人资料"}

- item\_form

- form{"name": "item\_form"}

- slot{"defendant": "张青红"}

- slot{"item": "个人资料"}

- form{"name": null}

- slot{"requested\_slot": null}

\* restart

- action\_restart

- utter\_answer\_restart

\* request\_info{"defendant": "小黑", "item": "个人资料"}

- item\_form

- form{"name": "item\_form"}

- slot{"defendant": "小黑"}

- slot{"item": "个人资料"}

- form{"name": null}

- slot{"requested\_slot": null}

\* affirm

- utter\_answer\_affirm

## interactive\_story\_23

\* greet

- utter\_answer\_greet

\* request\_info{"defendant": "小红", "item": "个人资料"}

- item\_form

- form{"name": "item\_form"}

- slot{"defendant": "小红"}

- slot{"item": "个人资料"}

- form{"name": null}

- slot{"requested\_slot": null}

\* inform{"defendant": "小明"}

- item\_form

- form{"name": "item\_form"}

- slot{"defendant": "小红"}

- slot{"item": "个人资料"}

- slot{"defendant": "小明"}

- form{"name": null}

- slot{"requested\_slot": null}

\* restart

- action\_restart

- utter\_answer\_restart

\* request\_info{"defendant": "陈远清", "item": "个人资料"}

- item\_form

- form{"name": "item\_form"}

- slot{"defendant": "陈远清"}

- slot{"item": "个人资料"}

- form{"name": null}

- slot{"requested\_slot": null}

\* inform{"item": "性别"}

- item\_form

- form{"name": "item\_form"}

- slot{"defendant": "陈远清"}

- slot{"item": "信息"}

- slot{"item": "性别"}

- form{"name": null}

- slot{"requested\_slot": null}

\* affirm

- utter\_answer\_affirm

## interactive\_story\_24

\* greet

- utter\_answer\_greet

\* request\_case

- case\_form

- form{"name": "case\_form"}

- slot{"requested\_slot": "case"}

\* form: inform{"case": "浙1125刑初148号"}

- form: case\_form

- slot{"case": "浙1125刑初148号"}

- slot{"case": "浙1125刑初148号"}

- form{"name": null}

- slot{"requested\_slot": null}

\* thanks

- utter\_noworries

## interactive\_story\_25

\* greet

- utter\_answer\_greet

\* request\_info

- utter\_ask\_name

\* request\_info{"defendant": "陈远清"}

- item\_form

- form{"name": "item\_form"}

- slot{"defendant": "陈远清"}

- slot{"requested\_slot": "item"}

\* form: inform{"item": "性别"}

- form: item\_form

- slot{"item": "性别"}

- form{"name": null}

- slot{"requested\_slot": null}

\* affirm

- utter\_answer\_affirm

## interactive\_story\_26

\* greet

- utter\_answer\_greet

\* request\_info{"defendant": "陈远清", "item": "个人资料"}

- item\_form

- form{"name": "item\_form"}

- slot{"defendant": "陈远清"}

- slot{"item": "个人资料"}

- form{"name": null}

- slot{"requested\_slot": null}

\* inform{"defendant": "小黑"}

- item\_form

- form{"name": "item\_form"}

- slot{"defendant": "陈远清"}

- slot{"item": "个人资料"}

- slot{"defendant": "小黑"}

- form{"name": null}

- slot{"requested\_slot": null}

\* thanks

- utter\_noworries

## interactive\_story\_27

\* greet

- utter\_answer\_greet

\* request\_info{"item": "性别"}

- item\_form

- form{"name": "item\_form"}

- slot{"item": "性别"}

- slot{"requested\_slot": "defendant"}

- utter\_ask\_name

- action\_listen

\* form: inform{"defendant": "小黑"}

- item\_form

- slot{"defendant": "小黑"}

- form{"name": null}

- slot{"requested\_slot": null}

\* restart

- action\_restart

- utter\_answer\_restart

\* inform{"defendant": "陈远清"}

- item\_form

- form{"name": "item\_form"}

- slot{"defendant": "陈远清"}

- slot{"requested\_slot": "item"}

\* form: inform{"item": "民族"}

- form: item\_form

- slot{"item": "民族"}

- form{"name": null}

- slot{"requested\_slot": null}

\* affirm

- utter\_answer\_affirm

\* thanks

- utter\_answer\_thanks

## interactive\_story\_28

\* greet

- utter\_answer\_greet

\* request\_info

- utter\_ask\_name

\* inform{"item": "个人资料"}

- item\_form

- form{"name": "item\_form"}

- slot{"item": "个人资料"}

- slot{"requested\_slot": "defendant"}

- utter\_ask\_name

- action\_listen

\* form: inform{"defendant": "陈远清"}

- item\_form

- slot{"defendant": "陈远清"}

- form{"name": null}

- slot{"requested\_slot": null}

\* affirm

- utter\_answer\_affirm

## interactive\_story\_229

\* greet

- utter\_answer\_greet

\* request\_info

- utter\_ask\_name

\* inform{"defendant": "陈远清"}

- item\_form

- form{"name": "item\_form"}

- slot{"defendant": "陈远清"}

- slot{"requested\_slot": "item"}

\* inform

- utter\_default

- action\_listen

\* form: inform{"item": "现住址"}

- item\_form

- slot{"item": "现住址"}

- form{"name": null}

- slot{"requested\_slot": null}

\* thanks

- utter\_noworries

## interactive\_story\_30

\* greet

- utter\_answer\_greet

\* whattodo

- utter\_answer\_whattodo

\* request\_info{"defendant": "陈远清", "item": "现住址"}

- item\_form

- form{"name": "item\_form"}

- slot{"defendant": "陈远清"}

- slot{"item": "现住址"}

- form{"name": null}

- slot{"requested\_slot": null}

【案例代码】actions.py

from typing import Any, Text, Dict, List,Union

from rasa\_sdk import Action, Tracker

from rasa\_sdk.executor import CollectingDispatcher

from rasa\_sdk.forms import FormAction

from rasa\_sdk.events import UserUtteranceReverted, Restarted, SlotSet

#

from py2neo import Graph, NodeMatcher

graph =Graph(

host="localhost", # neo4j 搭载服务器的ip地址，ifconfig可获取到

http\_port=7474, # neo4j 服务器监听的端口号

user="neo4j", # 数据库user name，如果没有更改过，应该是neo4j

password="!@#$1234")

selector = NodeMatcher(graph)

class CaseForm(FormAction):

"""Example of a custom form action"""

def name(self) -> Text:

"""Unique identifier of the form"""

return "case\_form"

@staticmethod

def required\_slots(tracker: Tracker) -> List[Text]:

"""A list of required slots that the form has to fill"""

return ["case"]

def slot\_mappings(self) -> Dict[Text, Union[Dict, List[Dict]]]:

"""A dictionary to map required slots to

- an extracted entity

- intent: value pairs

- a whole message

or a list of them, where a first match will be picked"""

return {

"case": [

self.from\_entity(

entity="case", intent=["inform", "request\_case"]

),

self.from\_entity(entity="case"),

]

}

def submit(

self,

dispatcher: CollectingDispatcher,

tracker: Tracker,

domain: Dict[Text, Any],

) -> List[Dict]:

"""Define what the form has to do

after all required slots are filled"""

case = tracker.get\_slot('case')

if not case:

dispatcher.utter\_message("服务器开小差了")

return []

all\_defendants = ""

defendants = list(selector.match("被告人", 案件号\_\_contains=case))

n = defendants.\_\_len\_\_()

if n==0:

response = "这个系统数据还够完善, 没有找到关于案件'{}'的信息, 抱歉哦..".format(case)

dispatcher.utter\_message(response)

return []

for ind in range(n):

if ind == n - 1:

all\_defendants = all\_defendants + defendants[ind]['name'] + "."

else:

all\_defendants = all\_defendants + defendants[ind]['name'] + ','

response = "{}案件, 有涉案人员:{}, 共{}人".format(case, all\_defendants, n)

dispatcher.utter\_message(response)

return [SlotSet('case', case)]

class ItemForm(FormAction):

"""Example of a custom form action"""

def name(self) -> Text:

"""Unique identifier of the form"""

return "item\_form"

@staticmethod

def required\_slots(tracker: Tracker) -> List[Text]:

"""A list of required slots that the form has to fill"""

return ["defendant","item"]

def slot\_mappings(self) -> Dict[Text, Union[Dict, List[Dict]]]:

"""A dictionary to map required slots to

- an extracted entity

- intent: value pairs

- a whole message

or a list of them, where a first match will be picked"""

return {

"defendant":[

self.from\_entity(

entity="defendant", intent=["inform", "request\_info"]

),

self.from\_entity(entity="defendant"),

],

"item": [

self.from\_entity(

entity="item", intent=["inform", "request\_info"]

),

self.from\_entity(entity="item"),

]

}

def submit(

self,

dispatcher: CollectingDispatcher,

tracker: Tracker,

domain: Dict[Text, Any],

) -> List[Dict]:

"""Define what the form has to do

after all required slots are filled"""

defendant = tracker.get\_slot('defendant')

item = tracker.get\_slot('item')

if (item == None or defendant == None):

dispatcher.utter\_message("服务器开小差了")

return []

response="这个系统数据还够完善, 没有找到{}关于'{}'的信息, 抱歉哦..".format(defendant, item)

person = graph.nodes.match("被告人", name=defendant).first()

if not person:

dispatcher.utter\_message(response)

return []

if item.find("个人资料") != -1:

response = "{},{},{}生,户籍所在:{}, {}程度, 现住{}, 贩毒{}".format(defendant, person['性别'], person['出生日期'],

person['户籍所在地'], person['文化程度'], person['现住址'],

person['毒品数量'])

elif item.find("出生地") != -1:

response = "{}的出生地是{}".format(defendant, person['出生地'])

elif item.find("生日") != -1:

response = "{}的生日是{}".format(defendant, person['出生日期'])

elif item.find("性别") != -1:

response = "{}的性别是:{}".format(defendant, person['性别'])

elif item.find("户籍所在地") != -1:

response = "{}的户籍所在地是:{}".format(defendant, person['户籍所在地'])

elif item.find("文化程度") != -1:

response = "{}的文化程度是:{}".format(defendant, person['文化程度'])

elif item.find("贩毒量") != -1:

response = "{}的贩毒量是:{}".format(defendant, person['毒品数量'])

elif item.find("民族") != -1:

response = "{}的民族是:{}".format(defendant, person['民族'])

elif item.find("现住址") != -1:

response = "{}的现住址是:{}".format(defendant, person['现住址'])

elif item.find("职业") != -1:

response = "{}的职业是:{}".format(defendant, person['职业'])

dispatcher.utter\_message(response)

return []

【案例代码】config.py

language: "zh"

pipeline:

- name: "JiebaTokenizer"

dictionary\_path: "data/dict"

- name: "RegexFeaturizer"

- name: "CRFEntityExtractor"

- name: "EntitySynonymMapper"

- name: "CountVectorsFeaturizer"

analyzer: "char\_wb"

min\_ngram: 1

max\_ngram: 4

- name: "EmbeddingIntentClassifier"

policies:

- name: TEDPolicy

epochs: 100

max\_history: 5

- name: MemoizationPolicy

max\_history: 5

- name: FormPolicy

- name: MappingPolicy

【案例代码】endpoints.yml

action\_endpoint:

url: "http://localhost:5055/webhook"