链滴

Rasa 笔记 1

作者：whitespur
原文链接：https://ld246.com/article/1548065503878
来源网站：链滴

通过本文可以建立起rasa的几个重要基本概念，rasa core,rasa nlu,story,domain,

intents　意图

actions　动作

templates　回答模板

entities　实体

slots　词槽

前言

本文内容基于Rasa官网文档，做了翻译与整理，并添加一些自己的理解，方便大家上手Rasa，想了更多内容的小伙伴可关注Rasa官网

Rasa是一个基于多轮对话的框架，其中包含两个模块Rasa core与Rasa nlu。

Rasa nlu是用来理解语义的，包括意图识别，实体识别，它会把用户的输入转换为结构化的数据，例在下图的例子中，nlu会识别出用户打算发一封邮件（意图），邮箱地址amy@example.com（实体。Rasa Core是一个对话管理的平台，它的工作是决定接下来机器该返回什么内容给用户，这里给用返回了Should we make that your primary email?

在接下来的内容中，我们会基于一个简单的多轮对话带领大家搭建一个Rasa平台，为了理解起来简单我们暂时不考虑填槽。

安装Rasa

pip install rasa_core

pip install rasa_nlu[tensorflow]

Rasa Core

我们先从Rasa core讲起，core包含两个内容，stories和domain。

1、Stories

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么样的，例如，在下文的子中，我们希望的流程是这样的：用户问好 -> 机器问用户今天过得怎么样 -> 用户反馈情绪 -> 机器据不同的情绪进行回复，这里其实包含两个流程，一个正面情绪的流程与一个负面情绪的流程，因此我们也需要编写两个story，接下来我们看下怎么编写story。

符号　说明

story 标题

- 意图

- 动作

story_happy

- greet
  - utter_greet
- mood_happy

- utter_happy

story_unhappy

- greet2
  - utter_greet
- mood_unhappy
- utter_unhappy

把以上内容保存到 stories.md文件中

2、Domain

domain可以理解为机器的知识库，其中定义了意图，动作，以及对应动作所反馈的内容。

标识　说明
intents　意图
actions　动作
templates　回答模板
entities　实体
slots　词槽
intents:

- greet
- mood_happy
- mood_unhappy

actions:

- utter_greet
- utter_happy
- utter_unhappy

templates:
utter_greet:

- text: "你好，你今天过的怎么样"

utter_happy:

- text: "那很棒棒哦"

utter_unhappy:

- text: "咋了，可以告诉我吗"

把以上内容保存到 domain.yml文件中

Rasa Core的任务是在获取到用户的意图后，选择正确的action，这些action就是定义在domain中以uter_开头的内容，每一个 action会根据templates中的情况来返回对应的内容。

在我们这个简单的例子中不需要定义词槽与实体，所以domain中暂时没有。

3、训练对话模型

下一步就是用神经网络去训练我们的Core模型了，我们可以直接执行以下命令，训练的模型将会存储models/dialogue文件夹下。

python -m rasa_core.train -d domain.yml -s stories.md -o models/dialogue

可以看到，训练过程采用了一个神经网络，结构为：masking->lstm->dense->activation，这里简介绍下masking层，在nlp领域，输入的内容可能不是一样长的，为了能统一处理数据需要定长，因某些值需要补0或者截取多余内容，但是补0的部分其实是没有意义的，masking层能让这些补0的部不参与之后的计算，从而提升运算效率。

4、尝试和你的机器人交流吧

接下来我们就可以用训练好的模型来运行我们的机器人了，执行以下命令

python -m rasa_core.run -d models/dialogue

此时我们的机器人还无法判断用户的意图，只能根据输入的意图返回特定的答案，所以我们只能输入些结构化的数据，例如输入我们之前在domian中定义好的意图，输入的信息需要以 /开头，我们可直接输入意图 /greet，当然，如果你想让机器人回答更多的内容，请在stories与domain中添加更多内容。

Rasa NLU

1、添加NLU模块

目前我们的机器人已经可以通过输入意图来获取答案了，但是怎么让机器理解真正的语言呢，这个时就需要用到NLU模块了，NLU的任务是解析消息，它能把自然语言解释成我们需要的结构化的数据，们继续完善下去。

首先，我们需要定义一个对应的意图可能会出现的文本内容文件nlu.md

intent:greet

- 你好
- 上午好
- 下午好
- 早上好
- 晚上好

intent:mood_happy

- 很好
- 我很好

intent:mood_unhappy

- 我很难受
- 我心情很差

把以上内容保存到 nlu.md文件中

除此之外，我们还需要一个nlu的配置文件，nlu_config.yml，由于我们是中文系统，所以language应的是zh，如果你需要英文的对话请修改为en。

language: zh

pipeline: tensorflow_embedding

把以上内容保存到nlu_config.yml文件中

准备好之后就可以开始训练NLU模型了，执行以下命令

python -m rasa_nlu.train -c nlu_config.yml --data nlu.md -o models --fixed_model_name nlu -project current --verbose

2. 再一次和你的机器人进行交流吧

添加完NLU模型之后我们就能让机器识别自然语言了，我们执行下以下命令。

python -m rasa_core.run -d models/dialogue -u models/current/nlu

到此我们简单的Rasa系统就搭建完成了，大家可以继续添加stories、domain、nlu文件的内容来搭一个属于自己的Rasa系统。下一篇文章将会带领大家学习如何做实体识别与填槽。

原文: https://terrifyzhao.github.io/2018/09/17/Rasa%E4%BD%BF%E7%94%A8%E6%8C%87%E%8D%9701.html