远程对话机器人项目总结报告

姓名：赵思蕊

目录

一、项目简介

项目是一个基于 Rasa_nlu 的对话机器人。它利用 rasa_nlu 的训练得到的 interpreter 分析意图，提取实体，得到需要的 params 后将 params 传递给 edamam api 查询菜谱，做出推荐。并可以进行简单的闲聊对话。

二、项目知识

1、正则表达式

正则表达式（英语：Regular Expression，常简写为 regex、regexp 或 RE），又称正则表示式、正则表示法、规则表达式、常规表示法，是计算机科学的一个概念。正则表达式使用单个字符串来描述、匹配一系列符合某个句法规则的字符串。在很多文本编辑器里，正则表达式通常被用来检索、替换那些符合某个模式的文本。

一个正则表达式通常被称为一个模式（pattern），为用来描述或者匹配一系列符合某个句法规则的字符串。例如：Handel、Händel 和 Haendel 这三个字符串，都可以由 H(a|ä|ae)ndel 这个模式来描述。大部分正则表达式的形式都有如下的结构：

选择[编辑]

(1)、竖线|代表选择（即或集），具有最低优先级。例如 gray|grey 可以匹配 grey 或 gray。

数量限定[编辑]

某个字符后的数量限定符用来限定前面这个字符允许出现的个数。最常见的数量限定符包括+、?和*（不加数量限定则代表出现一次且仅出现一次）：

(2)、加号+代表前面的字符必须至少出现一次。（1 次或多次）。例如，goo+gle 可以匹配 google、gooogle、goooogle 等;

(3)、问号?代表前面的字符最多只可以出现一次。（0 次或 1 次）。例如，colou?r 可以匹配 color 或者 colour;

(4)、星号*代表前面的字符可以不出现，也可以出现一次或者多次。（0 次、1 次或多次）。例如，0*42 可以匹配 42、042、0042、00042 等。

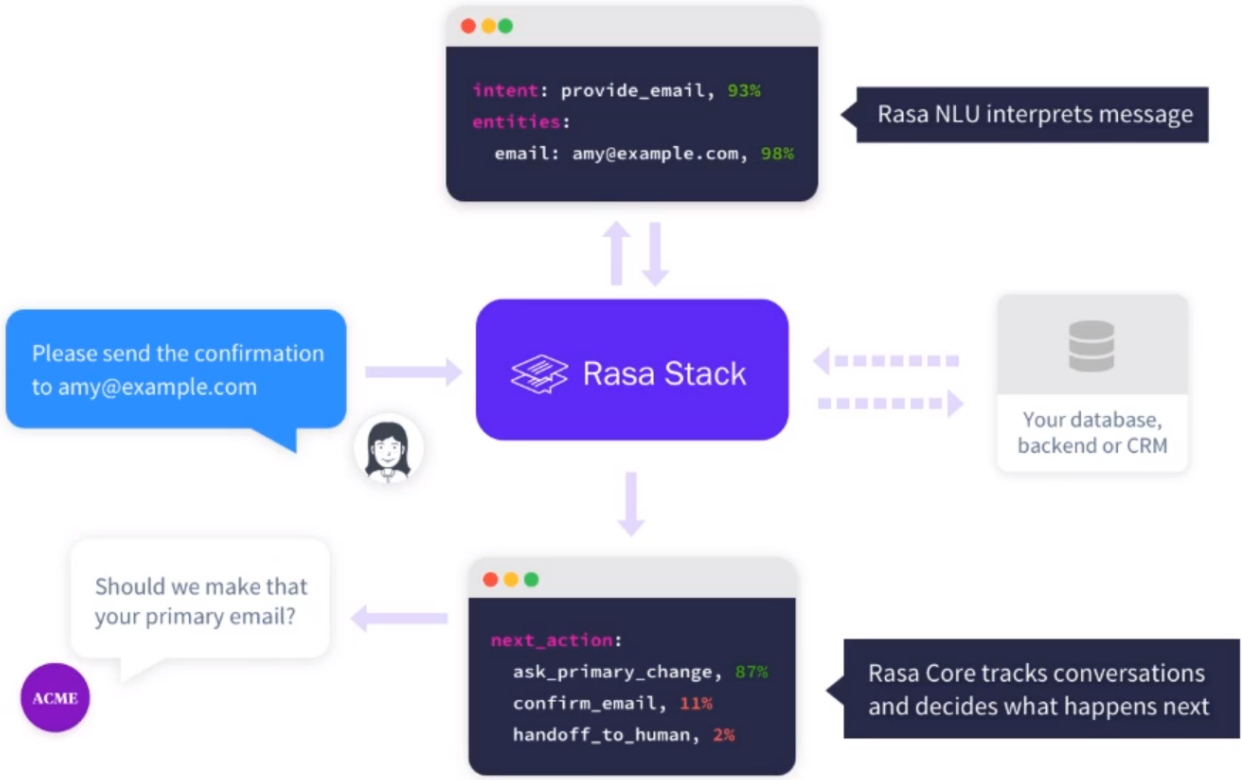
匹配[编辑]

(5)、圆括号()可以用来定义操作符的范围和优先度。例如，gr(a|e)y 等价于 gray|grey，(grand)?father 匹配 father 和 grandfather。

上述这些构造子都可以自由组合，因此 H(ae?|ä)ndel 和 H(a|ae|ä)ndel 是相同的，表示{"Handel", "Haendel", "Händel"}。

精确的语法可能因不同的工具或程序而异。

2、spaCy
spaCy 是一个 Python 自然语言处理工具包，诞生于 2014 年年中，号称“Industrial-Strength Natural Language Processing in Python”，是具有工业级强度的 Python NLP 工具包。spaCy 里大量使用了 Cython 来提高相关模块的性能，这个区别于学术性质更浓的 Python NLTK，因此具有了业界应用的实际价值。
它的特点有：非破坏性标记化、命名实体识别、“Alpha 标记化”支持超过 25 种语言、8 种语言的统计模型模型、预先训练的单词向量、词性标注、标记的依赖解析、语法驱动的句子分割、文字分类、用于语法和命名实体的内置可视化工具、深度学习整合
3、Rasa
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，例如在下图的例子中，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，这里给用户返回了 Should we make that your primary email?



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。
训练时需要的有 json 的 data 文件,和 yml 的配置文件。

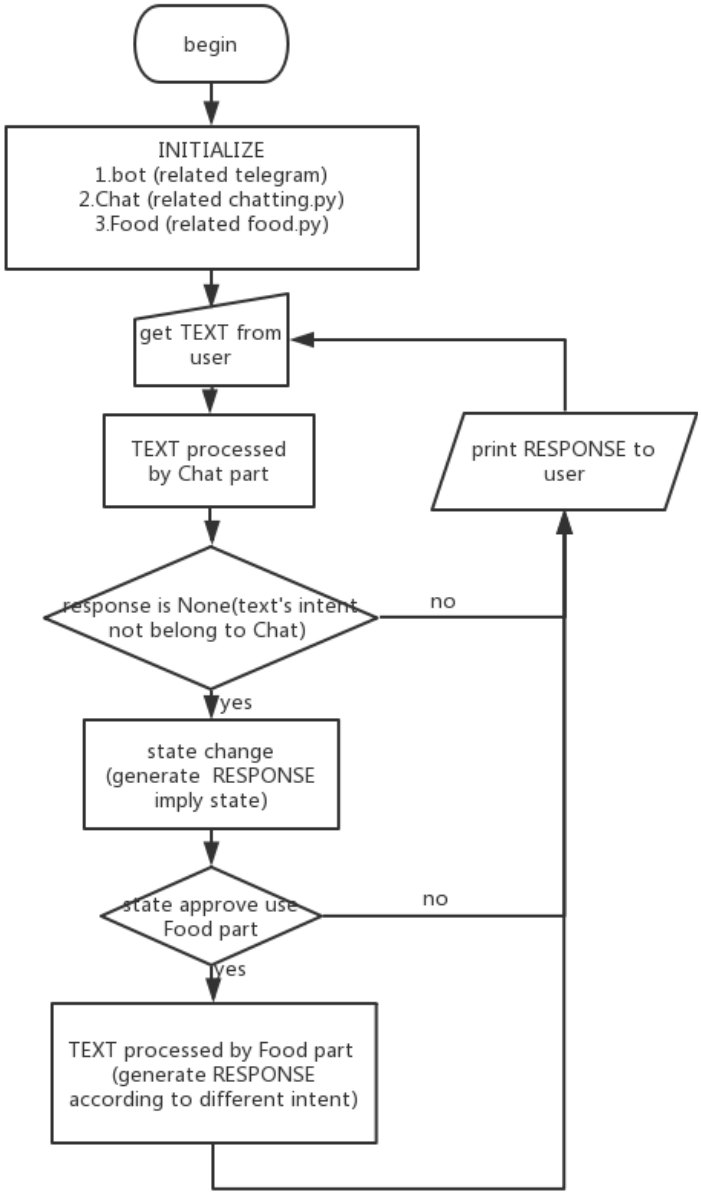
三、项目实现

1、基本思路

机器人的主要由 Chat,Food 两部分组成。

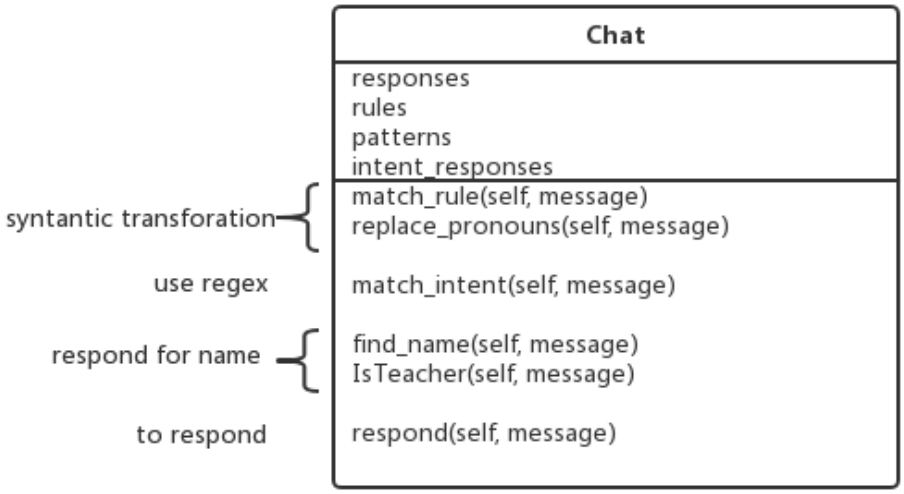
利用循环得到对话框的消息，把消息先利用 Chat part 处理，过滤属于闲聊的语句；再把不是闲聊的语句交付给 Food part 处理，这个部分里考虑到了状态的转变.

流程图如下：



2、Chat part

结合简单的选择性回答，文本替换，正则表达式，spacy 分析抽取名字实体等方式，实现简单的日常对话。



Chat 的 UML 类图

3、Food part

(1).考虑了状态的转变，当进入 RECIPE_SEARCH 后才可以利用 Food 的相关功能。

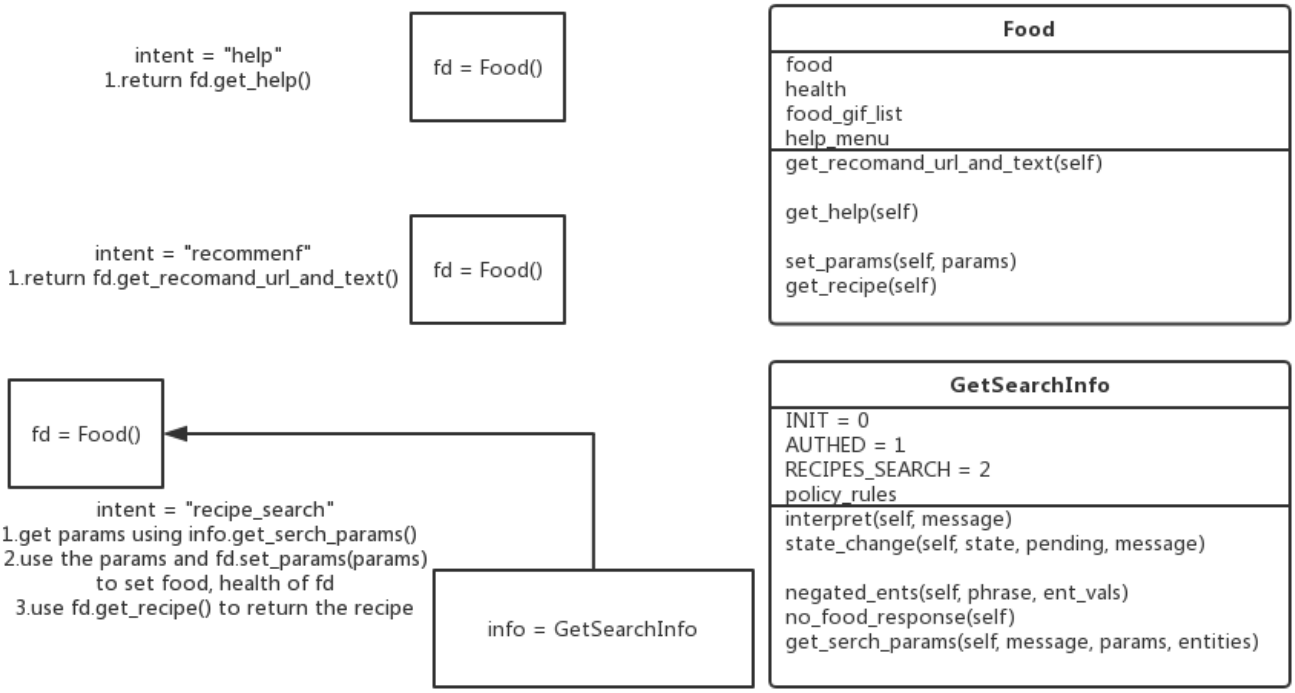
(2).进入 recipe_search 后，利用 rasa_nlu 训练好 interpreter 分析句子，得到句子的意图，抽取句子中的实体。训练好的模型存储在项目的 docs/rasa 目录下。

(3).Food 的功能有三项，也对应着 3 种意图，它们和它们的实现方式如下所示：

“help”:得到如何使用机器人的相关信息

“recommend”:推荐一些食物，简单的选择性回答,利用了有道翻译的 api 接口翻译了中文菜谱。

“search_recipe”：用 params 字典存贮搜索项的值，结合 interpreter 抽取的实体不断更新 params 参数，并加以多轮对话与否定功能。

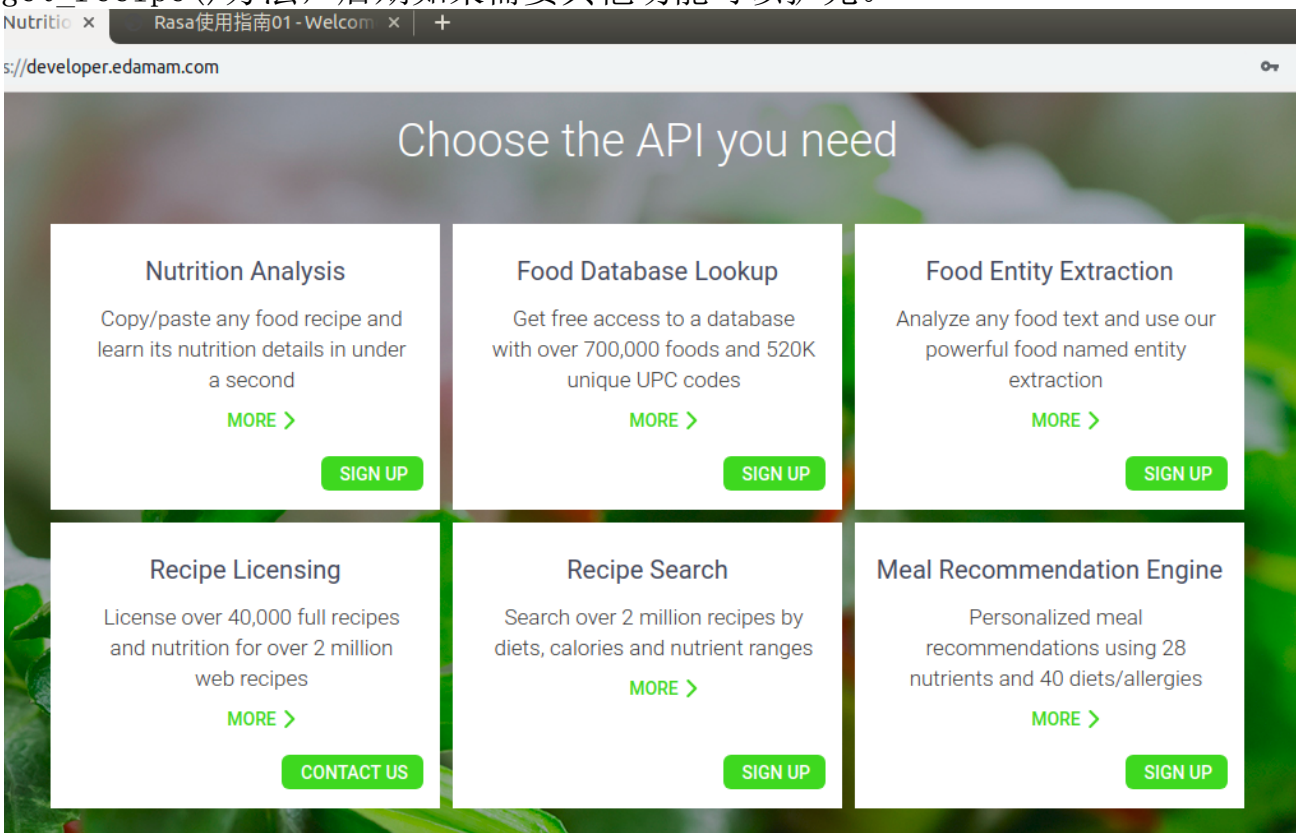


不同 intent 的处理方式 & Food、GetSearchInfo 的 UML 类图

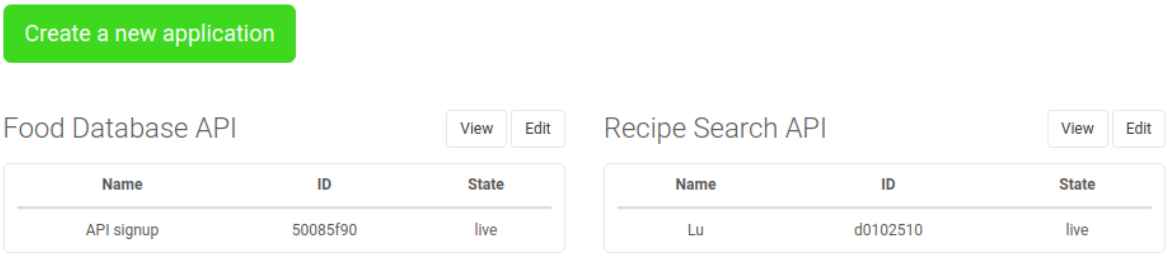
4、项目相关接口

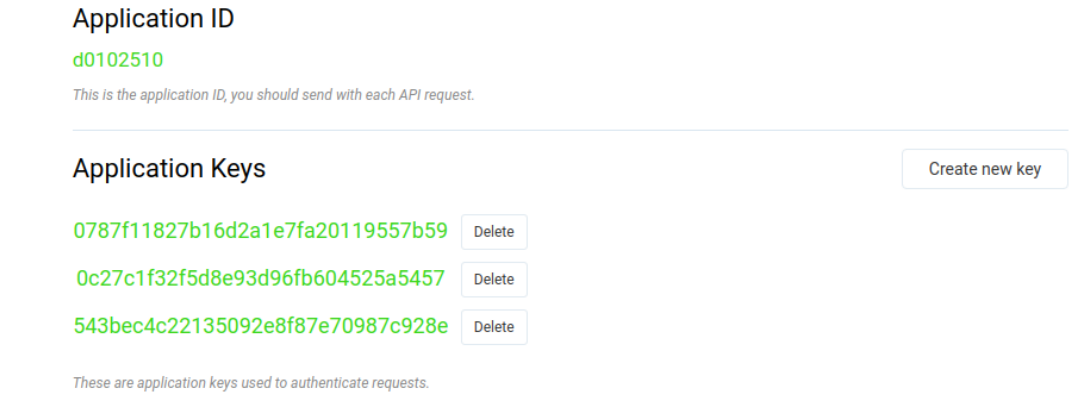
(1).Edamam:

有很多种类的 api。本项目中使用了它的 recipe search 部分实现 Food 类中的 get_recipe()方法，后期如果需要其他功能可以扩充。



选择想要的 api 应用，得到 Application ID 和 Application Keys.参考 Edamam 文档,利用 python requests 进行调用即可。



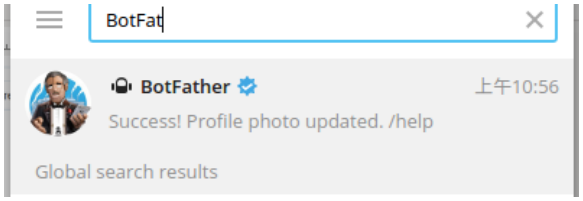


(2)利用了有道翻译接口编写了 Translate.py 将推荐部分的中文菜谱名称进行翻译。

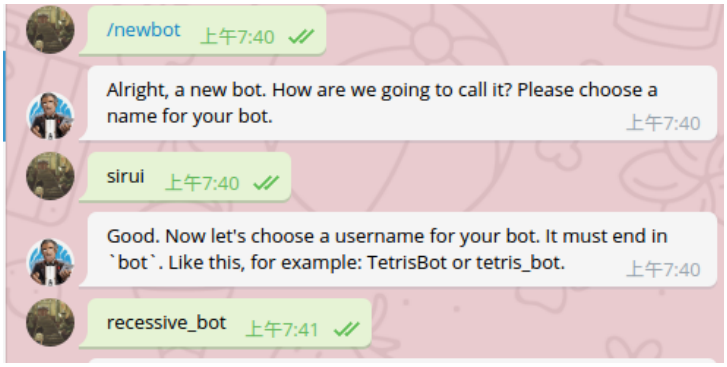
5、结果集成

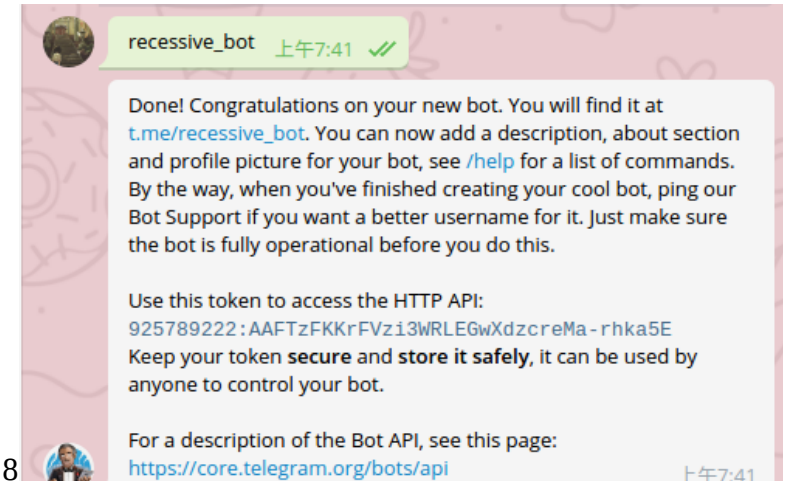
利用 Telegram 进行结果的集成。

搜索 BotFather 创建机器人：



按照提示创建机器人 BotFather 会返回给 Token 和相关信息，后面可以利用它在机器人和用户之间进行交互。





为了方便的发送信息，安装了 python-telegram-bot，它使得 telegram 相关 api 的调用更为简易。使用时 import telegram 即可。

6、机器人的设置

机器人的名字，和所有的 api,Token 都存贮在 robotConf.py 中

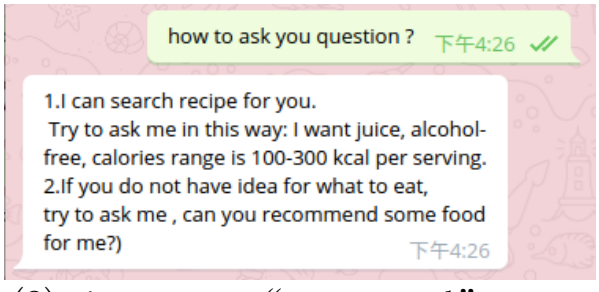
四、项目结果

1、聊天

2、状态转换

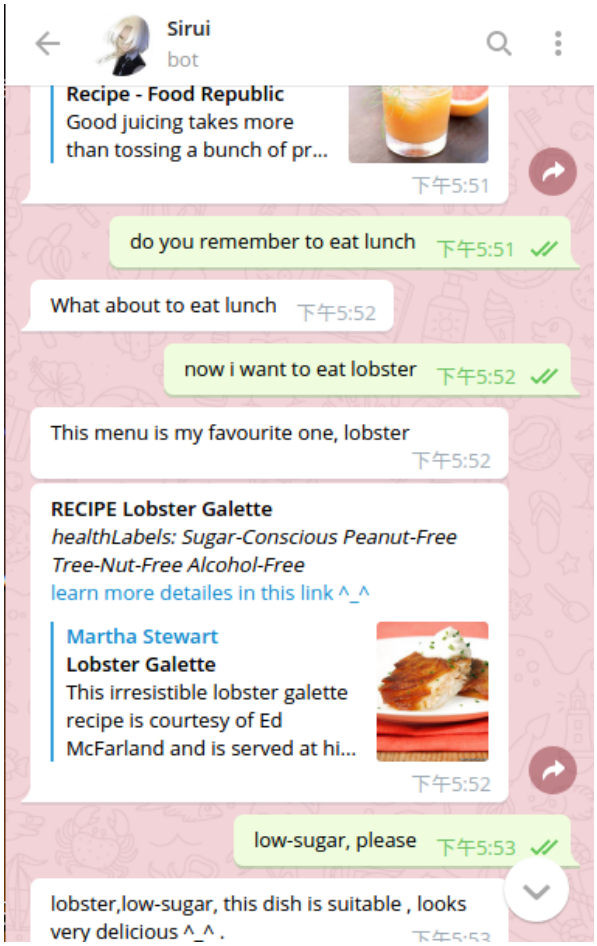
3、不同意图

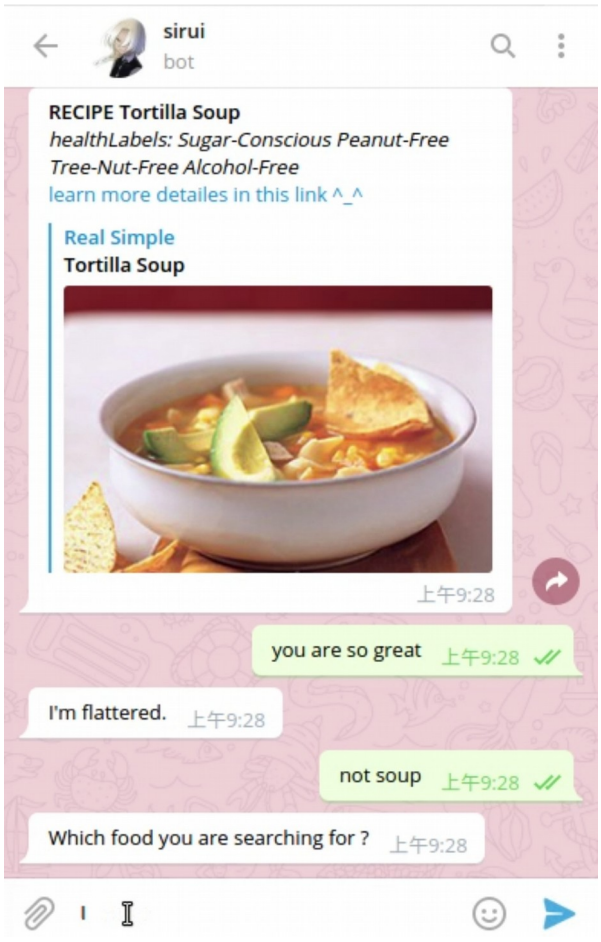
(1) intent = “help”



(2) intent = “recommend”

(3) intent = “search_recipe”
params :food, health
多轮对话，中途插入任意话语 进行否定





五、心得体会

为期一个月的机器人项目结束了。项目过程中学习了聊天机器人的九大技术，完成了项目美食推荐机器人，第一次有了完成整个项目的体验。项目的过

程中遇到了很多的困难和挑战，但在老师的引导之下，在不断着的探索之，最终完成。

项目里的不足

对时间的把握和项目完成安排上不合理。项目中途懈怠，项目后期加班加点。没有在一开始确定好目标计划，中途改道重做，精力时间的浪费。安排和计划事情的能力要加强锻炼，项目还有部分功能（搜索时添加卡路里取值范围）没有实现。

项目里获取对话框的信息是利用忙等待的方式测 bot.get_updates，比较低效，可以改进。

数据库的技术没有结合进来，可以考虑设计存储用户手机号码，个人信息，账户密码的数据库。

困难带来的启示

每一步几乎都在遇到问题，同样的步骤，运行结果却不同，甚至各种报错程序崩溃，仿若走入雷区。在困难打磨之下，从一开始，遇到问题就焦虑沮丧抱怨，转变为后来的，遇到错误觉得正常，努力排查错误继续进行。

Linux 系统的安装在几个小时还没成功的时候，就想到这个是最难的事了吧。然而，现实立刻用行动进行反驳。Anaconda 下了快 10 个版本，又因为 spacy 安装的不慎，重头再来。紧接着，下载 en_core_web_sm 等文件的时候，墙挡在了外面，于是翻墙。后来，微信集成遇到问题，不得不考虑其它的集成方式。在面对这些困难的过程中，老师给予了很多的帮助、鼓励、引导，很会调动人的积极性，有种只要是在计算机上的问题，老师总会有解决办法的感觉，而且答案一针见血，居然可以这样解决这个看起来无法解决的问题。

渐渐地项目好转了起来，在对待问题上也耐心起来。尽量不因为焦虑而耗费脑力，把它放到分析和解决问题上面。

对从前学习过程的反思

应该为自己选择方便喜欢适合的编程学习工作环境。安装配置好相关环境后，体会到了工具和环境的带来的便利，工欲善其事必先利其器。准备工作很重要。

应该提高主动性。长期以来自己不去主动学习了解计算机，不去动手实操。感叹坐井观天，画地为牢了这么久，墙外的世界很精彩， github 上有那么多优秀的代码，rasa 这样神奇的可以处理自然语言的模块 ……

参考

[1]Rasa 使用指南 1 https://terrifyzhao.github.io/2018/09/17/Rasa%E4%BD%BF%E7%94%A8%E6%8C%87%E5%8D%9701.html

[2]维基百科