1 聊天机器人设计需求分析

本项目所设计的聊天机器人是具有功能导向性的，它主要利用Movie Database (IMDB Alternative) API[4]来为用户提供关于一切影视作品的信息。通过借助于非常类似于互联网电影资料库（IMDB）[7]的Movie DataBase API[4]，聊天机器人可以帮助用户访问完整作品名、发行年份、发行日期、海报、作品类型、作品派系、情节介绍、演员、导演、作品时长、主要网站的评分、奖项等。

根据用户查阅影视作品数据的习惯，本项目机器人将需要如下工作部件。

首先，需要能够不断接受用户输入的自然语言消息，并分类处理消息的主模块。

其次，依据训练好的基于Rasa NLU的解释器对用户消息进行分析，识别消息意图并提取相关实体。

其三，根据意图做出不同行为反馈：如果意图包含检索目的，则根据提取到的实体连接API查找资源信息；如果是其他意图，直接做出相应消息回复。

其四，针对包含检索意图的消息，具体分析其所求，并利用槽填充技术返回相应的消息回复：机器人需要能根据用户提供的关键字获得所有条目，并通过一个将当前检索到的所有条目存储起来的范例，实现用户的二次直接访问某条目对应信息，而无需再次键入关键字的功能（即多次对话的实现）。

此外，机器人还应具备一定的闲聊功能，使其显得不呆板滞涩，而是能够较为生动地回复用户消息，增强用户体验。

最后，在机器人基本功能实现的基础上，通过将其集成在Telegram中来为用户提供一个可以有效访问和使用的平台。

2 智能机器人的设计与实现

根据上一步分析得出的机器人设计需求，这一部分将针对机器人的程序代码进行详细展开分析。程序代码是基于Python 3（根据张老师提供的源码素材）进行编写的，其中涉及的依赖库需要自行安装。

2.1 形成基于Rasa NLU的语料解释器

Rasa NLU是一种开源的自然语言处理工具，用于对话机器人中的意图分类、响应检索和实体提取等。它可以对一个普通自然语言句子进行分析并返回结构化数据。比如用户给出消息 “﻿help me find information about Avengers Endgame”，它将根据预先训练好的解释器返回结构化数据。

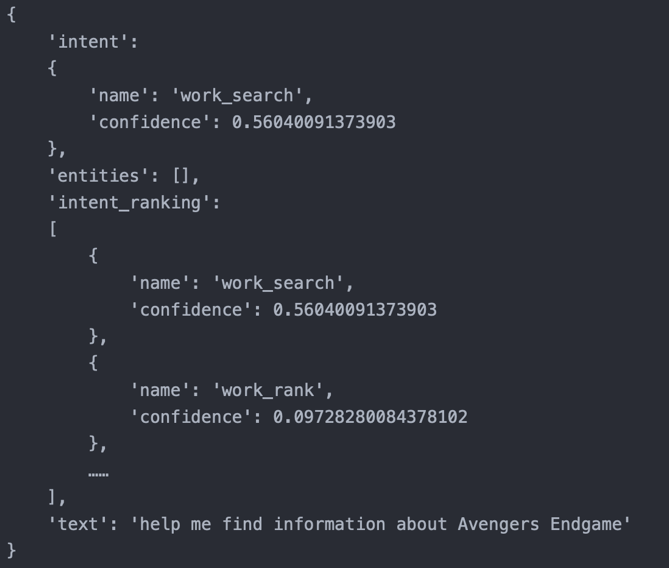
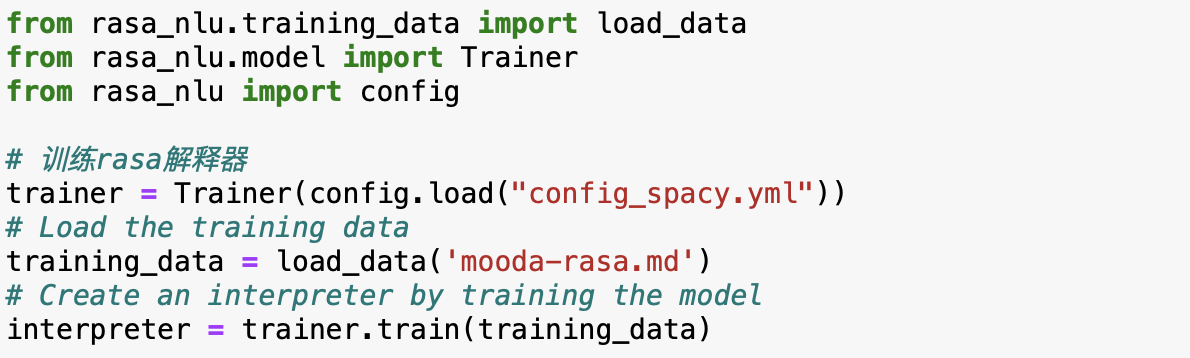


图 1 RASA NLU分析后的结构化数据

Fig.1 Structured data after RASA NLU analysis

想要实现解释器的功能，需要包含进来所有的依赖库，然后加载 .yml 后缀的配置文件和 .md 后缀的训练文件来生成解释器。关于 .md 格式的使用可以参考 Rasa 官网[3]：*[https://rasa.com/docs/](https://rasa.com/docs/ 。包含)* [。包含](https://rasa.com/docs/ 。包含) Rasa NLU 依赖库并生成解释器的代码如下。



其中配置文件中使用了语言包 “en\_core\_web\_md” ，采用 “spacy\_sklearn” 管道。训练文件所采用的.md（Markdown）格式是非常是和人们读写的格式，它可以使用[<entity text>](<entity name>)的语法来直接标示实体，比Json格式的文件简便。训练文件部分代码如下。



最后训练完成的解释器被存放在名为 “interpreter” 的变量中。使用“interpreter.parse(message)” 命令即可生成对消息（message）的解释。

2.2 提取用户消息中的电影关键字

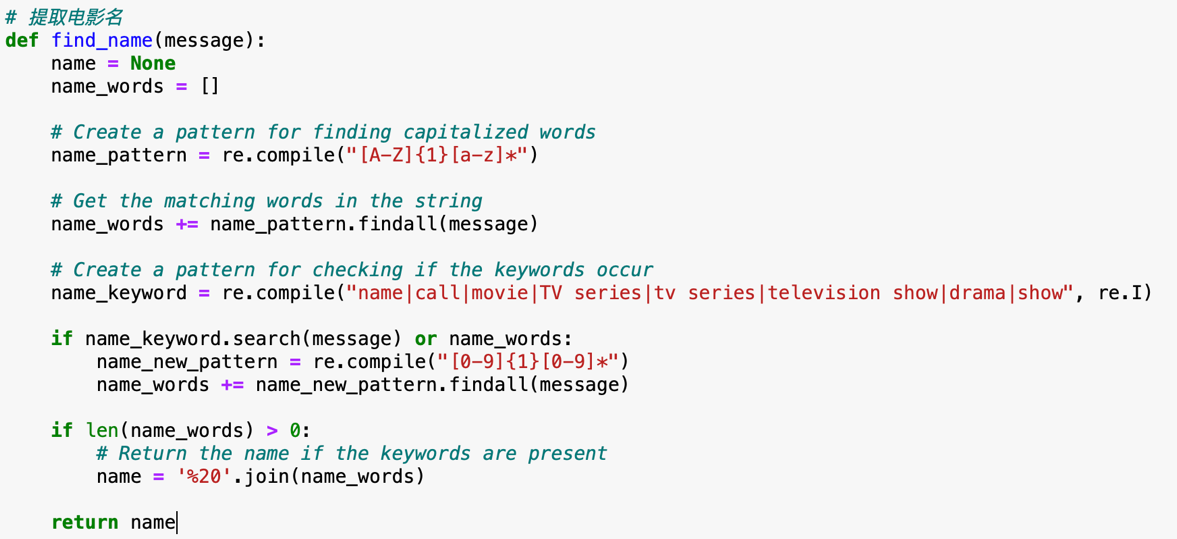
Rasa NLU虽然能够理解用户意图，但是对于电影名这种特殊的实体并不能很好的识别。由于 Rasa 是基于Spacy来实现的，如果想要验证 “en\_core\_web\_md”能有效提取哪些实体，可以在如下网站尝试：<https://explosion.ai/demos/>。

既然不能利用Rasa提取实体，我们可以转而利用原始的正则表达式方法来实现。为了使机器人能有效提取关键字，用户必须按照如下两种方式之一来键入电影名。

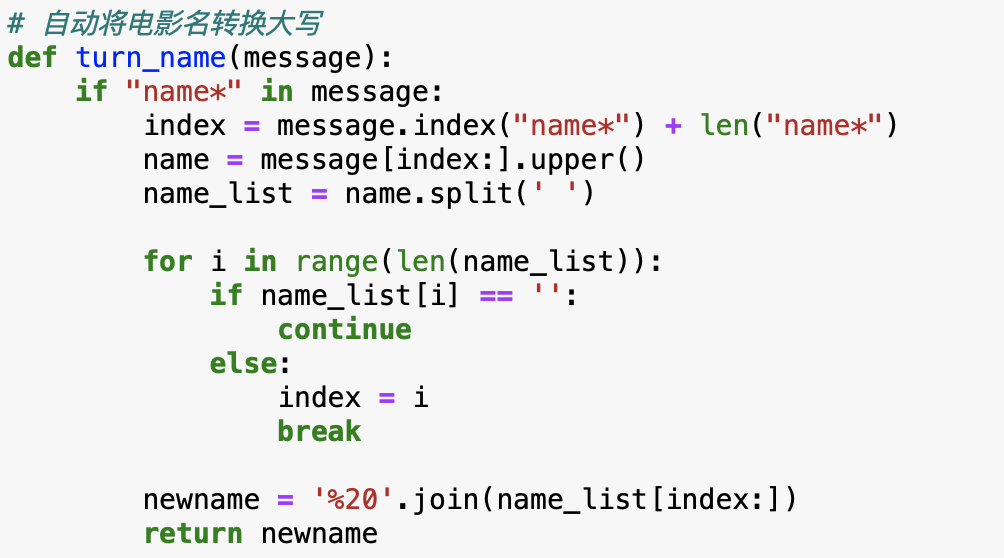
方式一，电影名每个单词首字母大写，比如输入 “i want information about Avengers: Endgame” 。

方式二，如果不想大写则需要在电影名前加上前缀 “name\*”，比如 “search name\* avengers: endgame” 。

实现首字母大写的电影名提取代码如下。



实现利用前缀 “name\*” 提取电影名的代码如下。

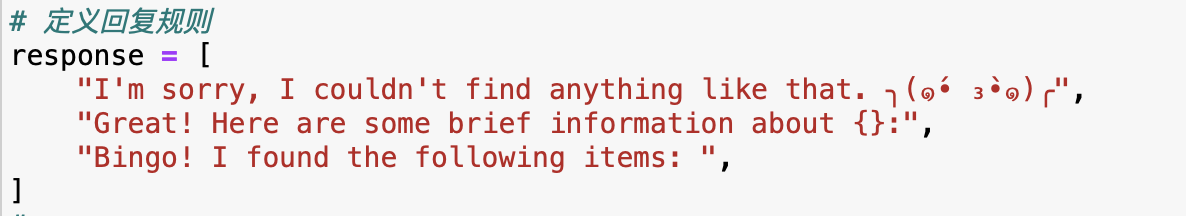


2.3 针对不同意图的后续响应

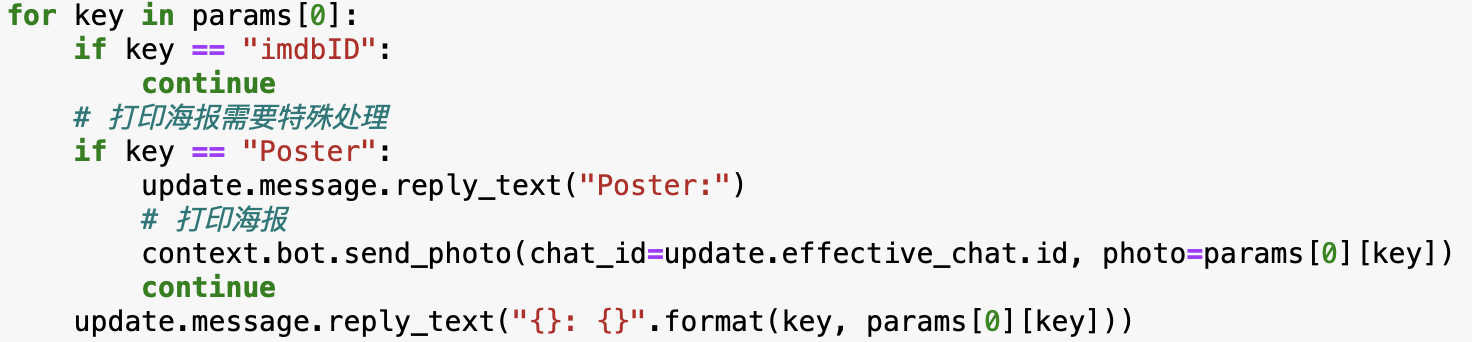
根据机器人需要实现的功能，我将检索意图做了如下细分，包括：搜索作品（work\_search）、给出作品索引（当搜索到多个条目）（work\_number）、搜索海报（work\_poster）、搜索发行日期（work\_date）、搜索作品类型（work\_genre）、搜索作品派系（work\_genre）、搜索情节介绍（work\_plot）、搜索演员名单（work\_actor）、搜索导演名单（work\_director）、搜索作品时长（work\_time）、搜索主要网站的评分和奖项（work\_rank）等。

当机器人合理理解了用户意图后，针对不同的意图，它应当适当调用Movie Database API来做出正确检索，根据检索内容进行合理回复，并根据检索到的条目生成范例以备后续多次对话使用。

针对搜索作品意图时，根据检索到的条目不同，需要给出不同的回答。如仅检索到一个条目，则直接给出该作品的简要介绍；如果检索出多个条目，则要提示用户选择一个条目；如果一个条目都没有检索到，则需要回复检索失败的信息。预定义的回答消息如下。



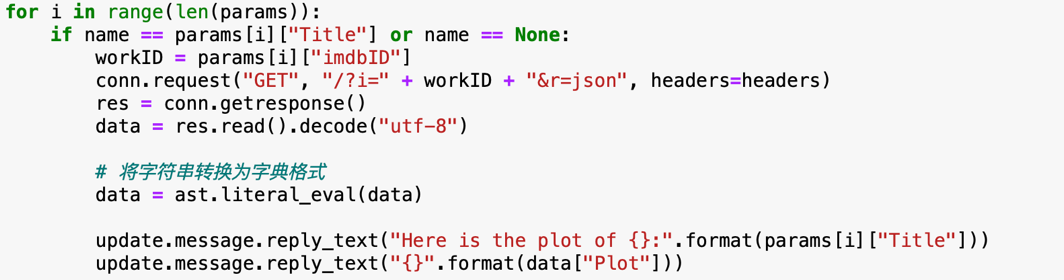
针对需要打印出一个作品简要信息的意图，比如搜索作品意图和给出作品索引意图时，需要做出不同于其他意图的反应。代码如下。



针对给出作品索引的意图时，为防止Rasa意图识别出错，采用了枚举的方式。即将一个数字（10以内）对应的所有表达列出，通过字符串匹配来确定索引。在索引的数字不存在时，给出提示重新输入的信息。代码略。

处理所有其他检索意图时，主要需要经过以下步骤：

1）检查所需条目是否已经存在于之前对话的范例中，并根据范例获得条目的IMDB ID号，已进行详细信息搜索。



2）如果条目不存在于范例中，则需要重新根据电影名检索，并自动回复检索到的第一个条目的相关信息。

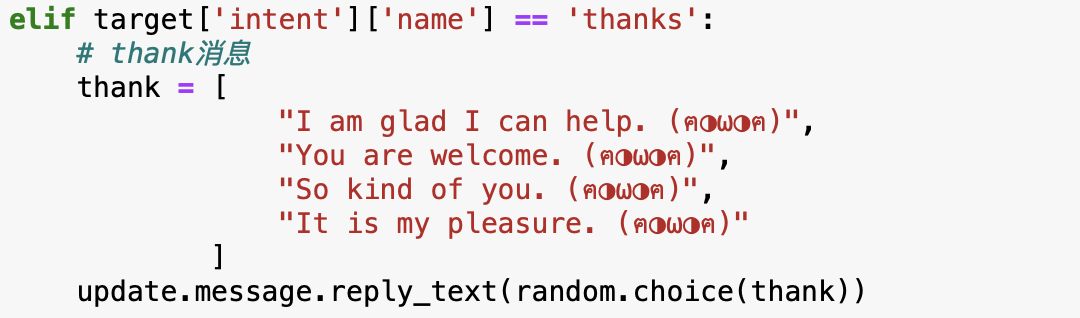


3）每次检索完毕都一定要“return params”更新范例（存储在params中）

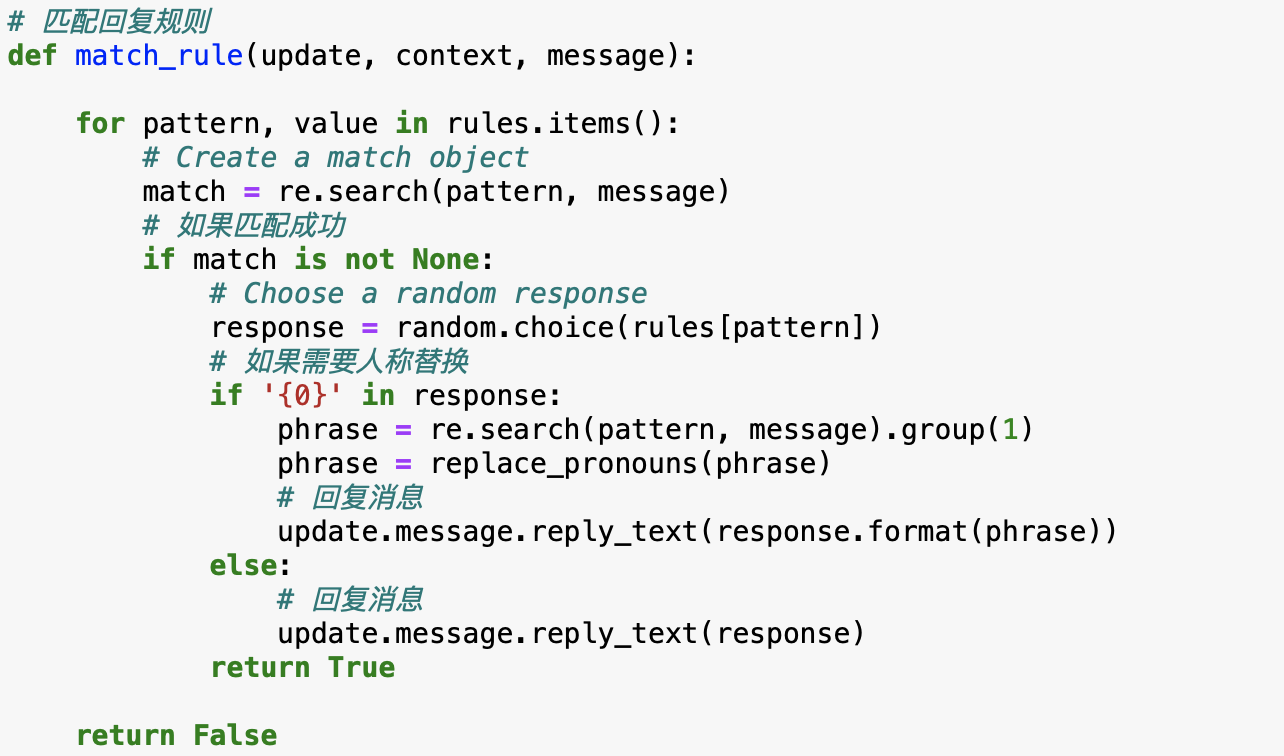
2.4 闲聊功能的实现

闲聊功能为增强聊天机器人的用户体验和趣味性意义非凡。本项目的机器人主要可以实现如下闲聊意图：问好、再见、质疑机器人身份、针对包含高兴情感消息的回复、用户答谢回复以及部分特定模式的闲聊回复（引用自课堂代码）。

其中所有闲聊意图都是根据Rasa NLU识别意图或者正则表达式匹配意图后，采用随机选择的方式从回复消息队列中选择回复的。比如针对用户答谢的回复代码如下。



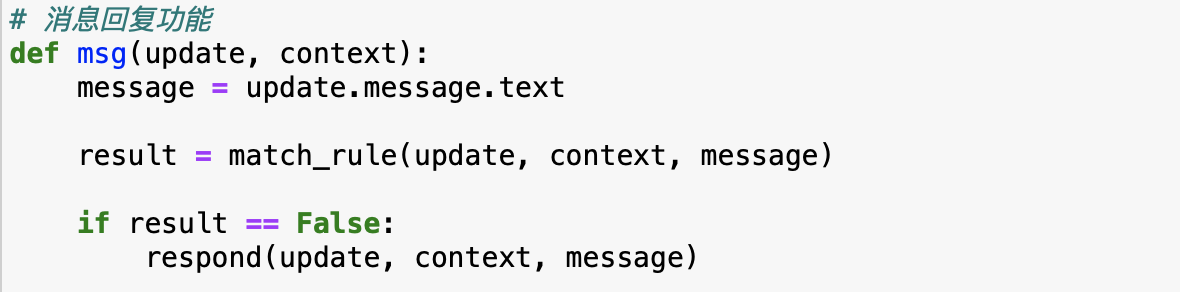
或者根据正则表达式匹配的回复如下（引用自课堂代码）。



2.5 处理用户消息的调度模块

调度模块主要实现对每一条用户消息进行意图识别并调用相关函数进行回复的功能。其中调度可以分为两个层次：根据是否闲聊进行调度；根据检索意图进行调度。

第一个层次，根据是否闲聊进行调度的代码如下。



第二个层次，根据检索目的的调度定义在respond函数中。通过 Rasa NLU识别的意图利用条件语句判断，实现跳转到其他函数的功能。其中判断意图部分过长，展示部分代码如下。



3 集成聊天机器人到Telegram

根据Telegram Bot API可以实现快速的对机器人的集成。本项目的集成过程主要分为如下步骤：

1)在Telegram上注册机器人；

2)为机器人创建相关对象体；

3)建立命令句柄实现在Telegram客户端对机器人的命令控制；

4)建立消息句柄实现接受和回复用户消息；启停控制。

Telegram Bot API依赖包的安装和使用详细细节可以参见官网：*<https://github.com/python-telegram-bot/python-telegram-bot>*，在此不做赘述。本项目集成的详细过程如下所示。

机器人注册可以直接在Telegram上与BotFather聊天来实现。输入 “/newbot”命令注册机器人，然后根据提示为机器人注册名字与用户名。本项目的机器人名为 “Mooda” ，用户名为 “mooda\_movie\_bot” 。用户可以通过这两种名字搜索到该机器人。

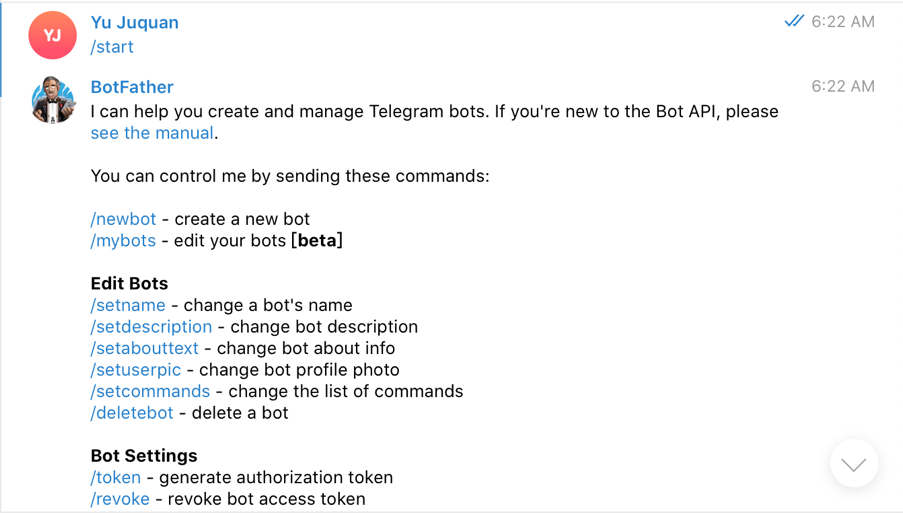
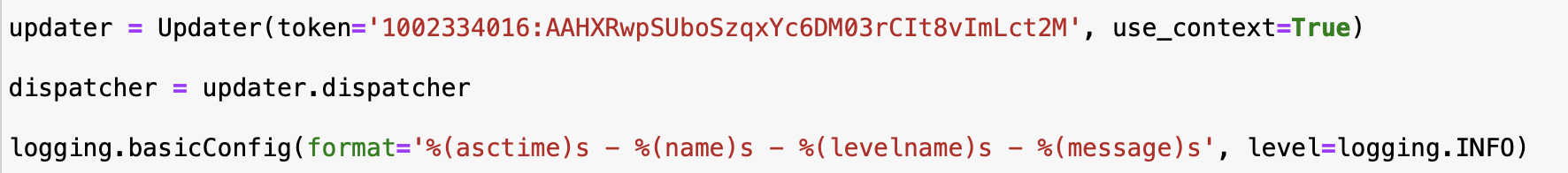


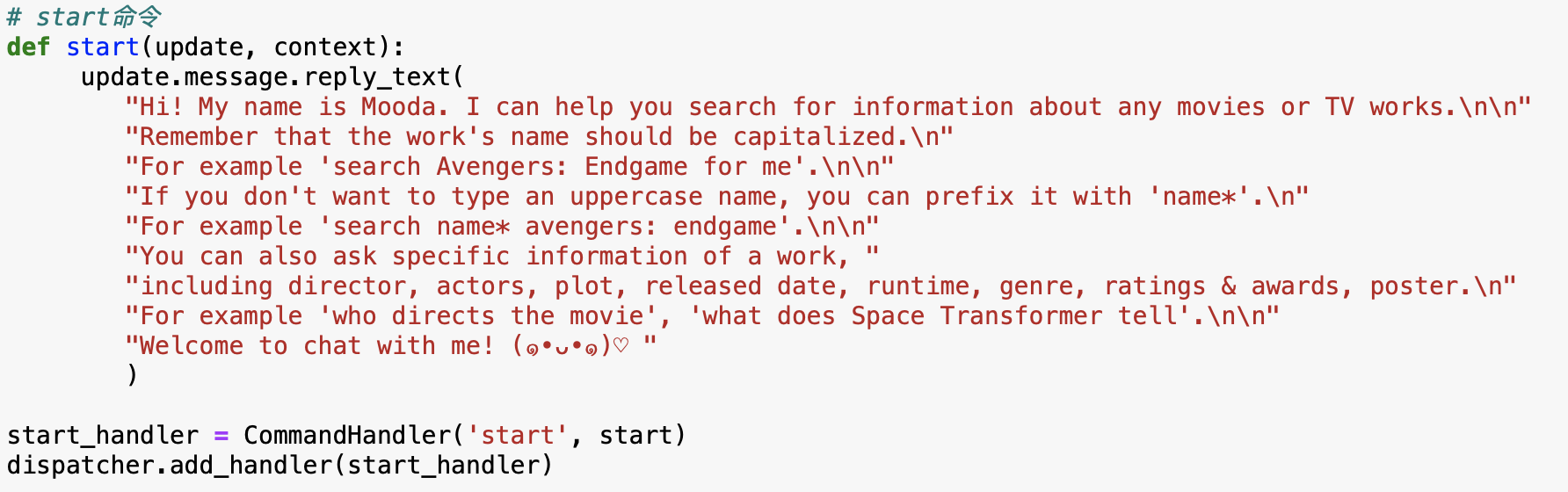
图 2 使用BotFather注册机器人

Fig.2 Registering robots with Botfather

然后根据获得的“token”来为机器人创建对象。



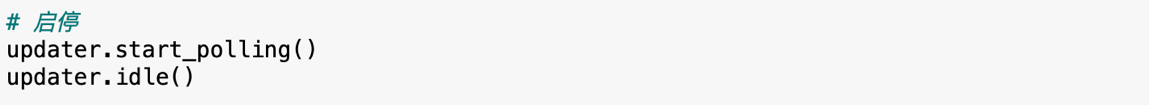
设置命令“/start”和命令句柄“start\_handler”。



设置消息句柄“msg\_handler”调用msg函数（msg函数已经在4.5部分展示）。



启停控制的设置。使用 idle() 后可以在命令行界面输入Ctrl + C来快速停止机器人的运行。



考虑到用户可能输入未知命令，设置 unknown\_handler 来对所有未知命令进行回复。



最后，启用内联模式（内联模式即用户输入@mooda\_movie\_bot即可调出Mooda的聊天界面）等工作，因为这些内容不是实现机器人所必须的，所以在此不做赘述。详细内容请参见Telegram Bot API官网或本项目源码。

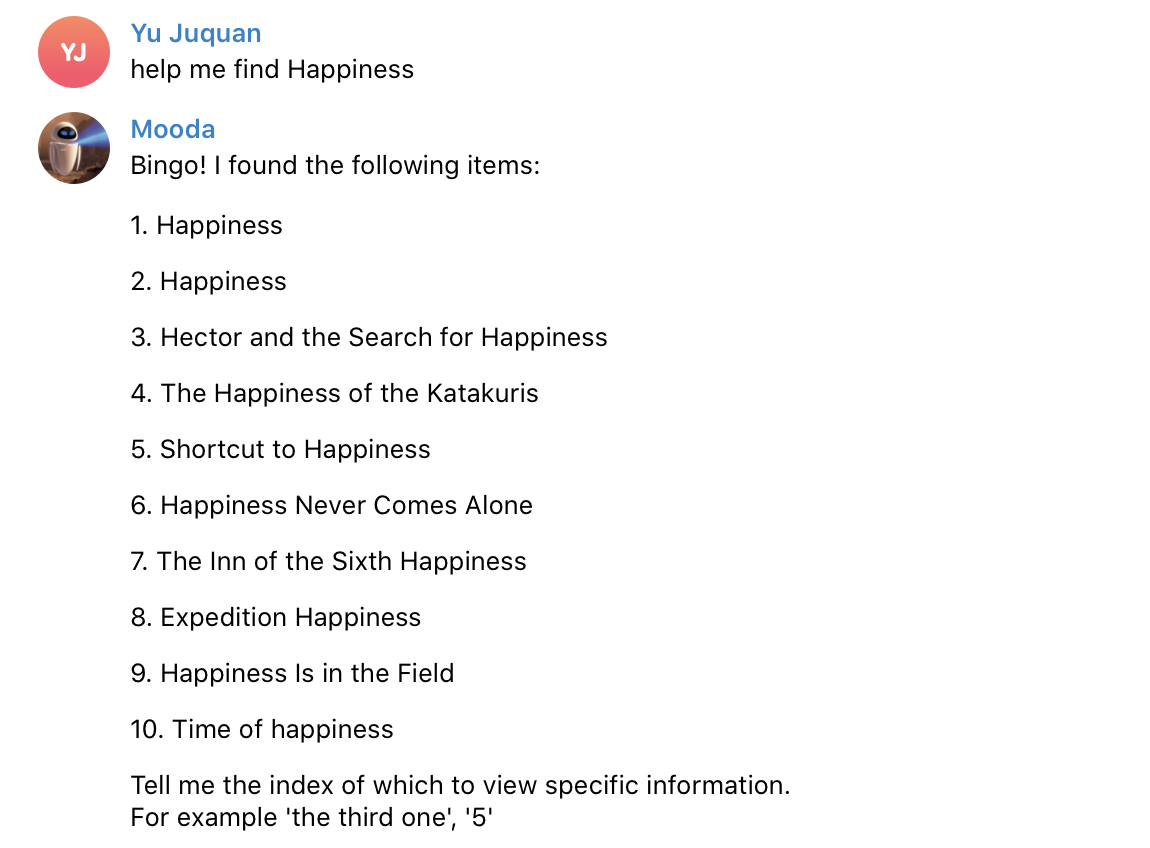
4 运行与测试

执行程序后，在Telegram界面打开Mooda聊天界面。输入 “/start” 命令后Mooda开始服务。

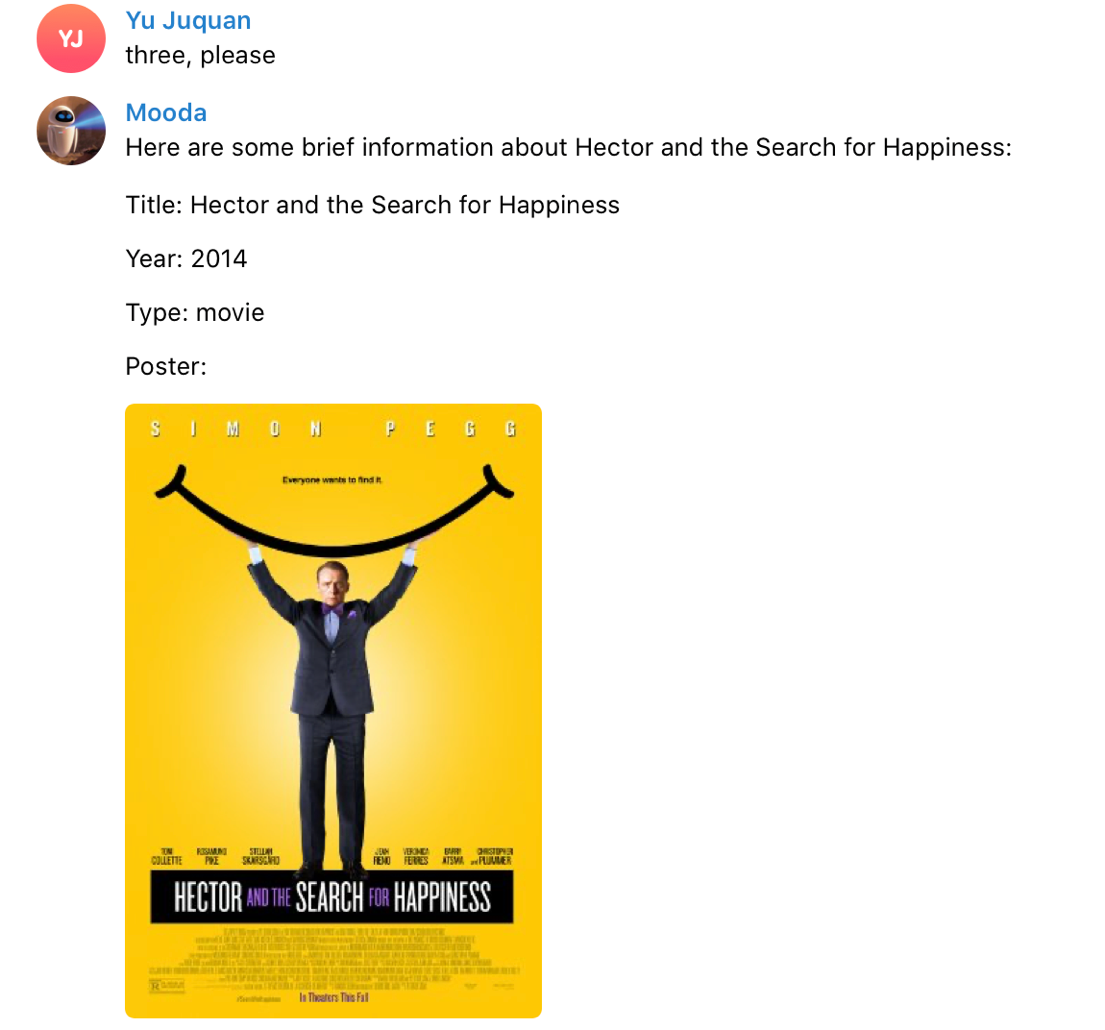
之后可以输入一系列包含不同意图的指令对机器人进行测试。

比如输入 “Hi” 同Mooda问好，会收到问候类回复如 “Hey~”

比如输入 “help me find Happiness”，会收到如下图所示的回复。



比如进一步输入 “three, please”，会得到第三个作品 “Hector and the Search for Happiness” 的简单介绍。



比如再进一步输入“what is it about”，会到的该作品 “Hector and the Search for Happiness” 的剧情简介。

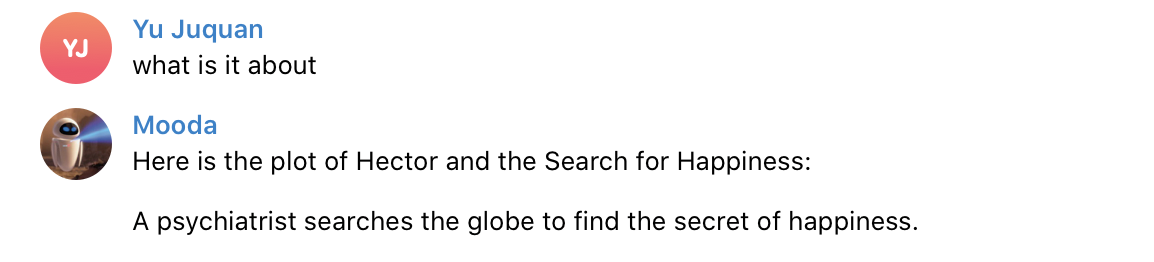


图 3-4-5 Mooda聊天界面

Fig.3-4-5 Mooda chat interface