远程科研指导项目——聊天机器人

蔡尔立

2019.12.06-2020.01.29

目录

1.研究背景

随着二十一世纪科技的快速发展，尤其是在近十年来，聊天机器人的前景和价值越来越得以显现。从商业前景来看，聊天机器人能够帮助企业提高运营效率，节约成本，为客户提供更多便利和增值服务，同时轻松解决各类客户问题，处理客户查询请求，降低人工交互需要。从生活方式上来看，自然语言作为人机交互界面，这无疑是比从鼠标键盘到触摸屏还要令人激动的巨大变革，彻底把各种智能设备的使用门槛降低为零。

关于聊天机器人的思考最早可以追溯到1950年图灵的论文《计算机器和智能》提出的疑问“机器人能思考吗”，图灵在论文中提出了图灵测试作为判定机器是否有智能的标准。关于图灵机器人的问题激发了MIT 的教授 Joseph Weizenbaum的兴趣，并于1966年开发了世界上第一款聊天机器人ELIZA，虽然可以和人进行简单对话，但更多的时候可能你看到最多的回复是“What are you saying about…”。 由于对话质量不尽如人意，以及应用场景的缺失，聊天机器人在过去的很长时间并未吸引太多的注意，仅仅是作为一项有趣的、半科幻的不太成熟的玩具存在着。

直到近十年来，随着深度学习技术的发展，围绕着聊天机器人的商业应用层出不穷，比如苹果的Siri、微软的Cnrtana和小冰、Google Now、IBM的Watson、亚马逊的蓝牙音箱等，不管是大企业还是小公司，都将聊天机器人看成是下一代人机交互的服务渠道。

从应用领域上来看，聊天机器人分为了两类，即开放领域和封闭领域。开放领域的聊天机器人可以和用户之间可以进行任何话题的自由对话，而封闭域问题通常有若干明确的目标和限定的知识范围。虽然开放域问题和图灵测试更接近，但也更困难。由于没有任何限定的主题或明确的目标，话题内容和形式更加的具有的不确定性，要准备的知识库和模型要复杂很多。相反的，封闭领域的聊天机器人所面临的输入和输出通常是有限的。虽然这个限定范围会随着问题领域以及对推理深度要求的不同变化很大，

但无论如何，与开放域问题相比，问题空间大大缩小，目标也更加清晰明确。

总的来说，虽然从实际的应用场景来看，**开放领域的聊天机器人能更广泛的应用在聊天、虚拟形象等泛娱乐领域**，用户基数比较大，也容易传播，但由于目的性不强、也会造成内容深度不够、对话质量不高等等一系列问题。而封闭领域的聊天机器人往往对对话错误的容忍度更低、对质量要求更高，这就要求聊天机器人对能够整合更多的领域知识、用户的基本信息，以及对上下文语境的分析和判断有很高的要求。

2016年，随着行业巨头微软、Facebook、亚马逊、Google 和苹果纷纷发布了各自在聊天机器人领域的战略和相关产品。聊天机器人无疑已经成为互联网业界和投资领域的热点之一。

2.知识点回顾

想要聊天机器人能对用户的问题作出相应的答复，而不是答非所问，最重要的就是能够提取用户的意图和句子的实体。

2.1 提取用户意图

本次项目组中提取用户的意图的方法，我们总共学到了三种，分别是使用正则表达式、最近邻分类法以及支持向量机。

其中，使用正则表达式的方法最为直观，我们首先需要建立一个意图与关键词相对应的字典，之后通过正则表达式搜索语句中的关键词来判断对应的意图。该方法简单且耗时短，但十分依赖于程序员对于具体问题的了解，需要能给出合适的关键词。

最近邻分类法则需要首先给出一个训练数据集并需要用到spaCy模组。spaCy模组能够对训练集中的每个句子以及需要判断语意的句子生成一个300维的向量，之后对目标语句和训练集中的每个句子求向量积，向量积越大，则说明两个句子的语意越相似，最后取向量积最大的句子（也

可以取向量积最大的几个句子）的语意为目标的语意。在训练集足够大的情况下该方法能在保证一定的正确率的情况小判断出句子的语意。

支持向量机法可以看做是对最近邻法的一个改进。在最近邻分类法中我们需要对训练集中的每个句子进行求向量积，可能会消耗大量的时间和运算资源。而支持向量机法中，取而代之的是首先对每种语意的向量求出一个中间值，最后将目标语句的向量和各个语意的向量进行比较即可。

这三种方法各有优劣，我在本次的项目中同时采取了正则表达式和支持向量机的方法。对于简单地语意，如问候、表示感谢以及再见，我采取了正则表达式的方法，因为这些语意可以通过句子中的一两个词来判断，拿表示感谢来举个例子，我们仅仅需要查询句子中是否有'thank'或者'gratitude'等词出现即可。而对于比较难的语意如需要查询某个歌手的一首歌，则需要通过rasa_nlu训练数据来分析其语意。

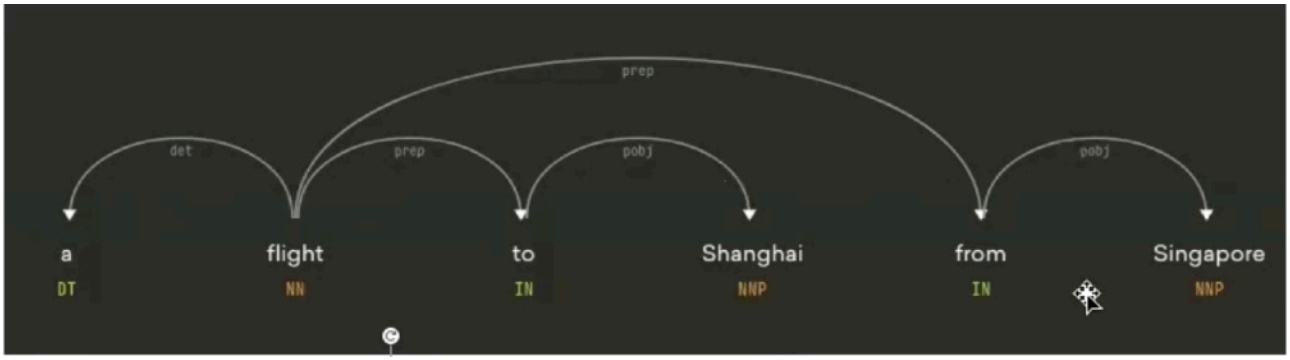
2.2 抽取实体

在抽取实体的方法上我们同样学习了三种方法，分别是通过预建的命名实体识别、角色关系、依赖分析来抽取实体。

预建的命名实体识别是通过使用spaCy自带的语料库来抽取实体，并且可以将实体分类到相应的标签中（date，orgnisation，person等）。

通过角色关系的抽取实体的方法是使用来抽取实体，在使用正则表达式时搜索类似"from Shang to Singapore"的句子，这么做的好处时能够保留实体之间的关系。

依赖分析的方法（如下图所示）则是在抽取实体时同时记录了每个实体的祖先，这个方法同样需要用到模块spaCy来对句子的词法，语法来



进行分析，这样做同样可以保留词与词之间的关联关系，以便之后可能要用到。

在得到了用户的意图之后，下一步要做到就是，对用户所提出的问题作出相应的回答，回答的内容可以从预先建立好的数据库中查找，但由于事先准备的数据不可能做到尽善尽美，也需要相应API来实现查询。

2.3 数据库的使用

本次项目中我们所使用的SQLite 是一款轻型的数据库，它的设计目标是嵌入式的，它有占用资源非常的低的优点，而且可以使用接近自然语言的代码来提取参数（如图所示）。

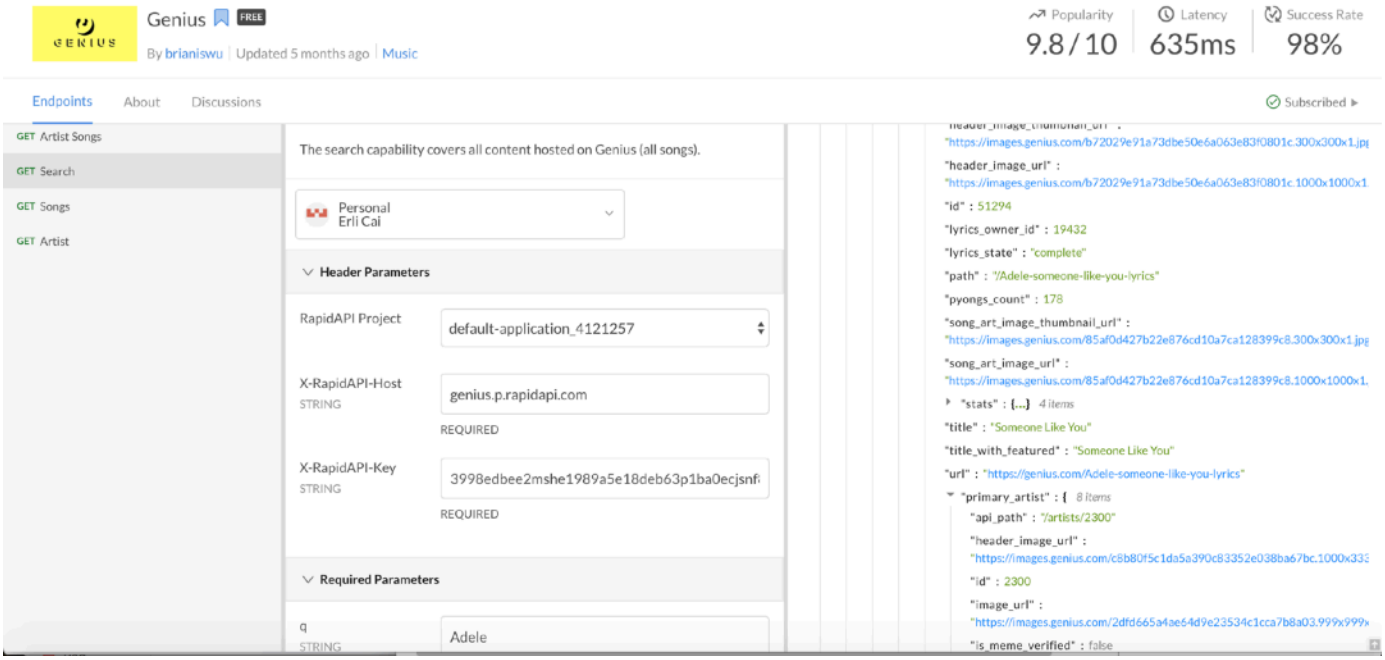
```
SELECT * from restaurants;

SELECT name, stars from restaurants;

SELECT name from restaurants WHERE location = 'center' AND price = 'hi';

SELECT name from restaurants WHERE location = 'center' AND price = 'mid';
```

2.4 API 的使用

显然，光使用本地简历的数据库是不能满足用户的需求的，这时我们就需要求助于第三方API（Application Programming Interface，应用程序编程接口）。简单地来说，使用API就是使用对方提供的服务，由于许多存在的问题都有了第三方解决方案，使用现有的解决方案已经变得更加方便且可靠了。一般，提供API的公司会建立专用的URL通道，用来返回纯数据响应，也就是说，响应内容不会包含图形用户界面（例如网站）中的显示开销，我们能通过这种方式轻松获得想要的数据。

本次的项目中，我使用的是由RapidAPI提供的genius API（下图战士了geniusAPI的界面，左侧为搜索的类别，中间需要填入搜索的人名/歌名，右侧则是搜索的结果），它能帮我查询到Genius（一个有着丰富歌曲

信息和歌词解析的网站）上如歌手的代表作，歌手的基本信息以及歌词等信息，而需要输入的信息仅仅只有歌手的名字。



2.5 多轮多次查询技术

对话机器人按场景可以分为多轮对话和单轮对话。单轮对话中只需考虑用户在本次对话中所表达的意思，而多轮对话是用户带着特定目的而来，需要在多次对话中满足特定限制条件的信息或服务，这使得我么需要记录当前的对话进行到了哪一步以及一些之前提到的参数，这种技术的实现主要依赖于对在回复消息的函数上加入一个状态参数来记录当前的状态 。

2.6 使用的模组

在本次的项目中多次用到了spacy和rasa_nlu这两个模组。

其中Rasa NLU 是一个开源的、可本地部署并配套有语料标注工具的自然语言理解框架。可以说自然语言理解（NLU）系统是聊天机器人的基石，尽管使用正则的方式也能在很多场景下实现一个简单的问答系统，但却缺乏泛化和学习的能力，而基于机器学习的 NLU 系统则可以举一反三不断学习改进模型从而获得越来越好的性能表现。

而spaCy模组则是一个Python自然语言处理工具包,它能够运用在对自然语言文本做词性分析、命名实体识别、依赖关系刻画，以及词嵌入向量的计算和可视化等方面

值得一提的是本次使用的rasa_nlu的版本为0.13.7，它并不能和3.7以及之后版本的python兼容，会产生如下的报错

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
<ipython-input-14-8ae745b543ae> in <module>
     12
     13 # Create an interpreter by training the model
---> 14 interpreter = trainer.train(training_data)

~/anaconda3/lib/python3.7/site-packages/rasa/nlu/model.py in train(self, data, **kwargs)
    189             logger.info(f"Starting to train component {component.name}")
    190             component.prepare_partial_processing(self.pipeline[:i], context)
--> 191             updates = component.train(working_data, self.config, **context)
    192             logger.info("Finished training component.")
    193             if updates:

~/anaconda3/lib/python3.7/site-packages/rasa/nlu/utils/spacy_utils.py in train(self, training_data, config, **kwargs)
    222
    223             for idx, example in enumerate(training_data.training_examples):
--> 224                 example_attribute_doc = attribute_docs[attribute][idx]
    225                 if len(example_attribute_doc):
    226                     # If length is 0, that means the initial text feature was None and was replaced by ''

IndexError: list index out of range
```

3. 心得体会

在这次项目的一个多月学习过程中我收获了很多，从最开始时使用python的不熟练，到看书学习相应语法以及模块的运用的知识，再到最后能独立的完成整个项目。这个过程中我不仅学到了许多关于聊天机器人的技术与知识，也了解到了行业的前景与最前沿的问题。最重要的是，本次的科研项目激发了我对于该领域的兴趣，可能和其它很多同学不同的是，我之前学习的并不是计算机专业，对于计算机方面的知识也十分有限，在参加这个项目之前，从未想过自己能独立的如此复杂项目。但有了这次的经历，相信我在今后的学习中遇到困难时也能够从容的去面对。