远程科研总结报告

人工智能方向：聊天机器人

姓名：XXX

学校：XXXXXXXXXX

专业：计算机科学与技术

时间：2019/9/24

目 录

1 项目背景与介绍

随着人工智能与自然语言处理技术的蓬勃发展，各种各样功能各异的聊天机器人一步步进入了我们的眼帘。从 IBM 的问答机器人 Watson 在电视问答节目中一战成名，到如今广泛用于餐饮点单、出行订票、售后问询的无人助手，再到智能语音助手 Bixby、Cortana 等，聊天机器人已经逐步走入并参与我们的生活。

而本次科研正是旨在使用自然语言处理的相关技术，设计、实现并部署一个具有实用功能的上下文智能聊天机器人。选题为音乐机器人，除了与用户进行常规闲聊，还能在对话中对用户意图和用于中的实体信息进行识别，根据用户要求，向用户推荐音乐、并在反馈中修改推荐结果，同时可根据录入的语音信息，进行听歌识曲。

2 项目技术简析

聊天机器人从建议到复杂，各式各样种类繁多，使用的相关技术也不尽相同。

早期，无需过多技术便可开发设计回音机器人与既定 FAQ 机器人。然而，回音机器人只是靠简单转述聊天者话语维持聊天，方式简单粗暴，并不需要对文本进行额外处理；既定 FAQ 聊天机器人只针对用户既定范围内的问题机型回答，否则机器人无法有效回复。即使大量扩充 FAQ 机器人问题库、丰富其回答句式数目，也只是一定程度内增加有效回复率和回答的丰富程度。

本次科研为能使机器人能获取人类话语的确切含义与意图，并根据意图有逻辑地与人聊天对话，还借助了其他技术工具：

2.1 聊天机器人基本技术与工具

2.1.1 正则表达式

正则表达式是简单却强大的文本处理工具。一个正则表达式并不是某种编程语言，而是一种文本模式，通常被用来描述、匹配、检索、操作那些符合某个正则表达式文本模式的字符串，很多高级程序语言都支持正则表达式操作字符串，此次项目使用的 Python 自然不例外。

正则表达式包含普通字符和特殊字符（又名元字符）。一个正则表达式表示的文本模式常可以匹配很多符合模式的字符串，这大大减轻了文本操作的繁琐程度。例如，正则表达式 go{2,}gle 可匹配所有含有 google、goooogle、goooogle 等等字符串。只要中间的 o 个数大于等于 2 个，就可以匹配成功。

正则表达式规则看似略显繁琐，但是功能非常强大，简化不少串操作。

2.1.2 文本嵌入与 spaCy 自然语言处理框架

人类语言中的词和句在计算机前，只是一个个不同的字符串。有些词语或句子虽然长得不同，含义却是一致的，而有些则毫不相关。如何让计算机表示与理解这些词语和的语义信息是一个问题。

向量模型给出了答案。初向量模型是基于统计学的方法（共现矩阵、SVD 分解），如今由基于不同结构的神经网络训练得到的一套可将文本映射为高维度的数字向量的模型，能把文本信息数字化，分为词向量和句向量模型。

在文本信息映射为多维空间内的高维向量后，向量两两之间的夹角的余弦值代表了他们原始文本信息的相似程度——两向量夹角越小，余弦越大，在空间内的方向越一致，原始文本信息越接近。

SpaCy 是一个强大的深度学习集成的自然语言处理框架，支持多种语言，可轻松完成各种 NLP 任务。本次项目采用 spaCy 作为自然语言处理工具，具体用途在后文呈现。

2.2 意图识别

前面提及的简易聊天 bot 中，机器人只是按照固定的简单规则，对用户进行简单的回答，但事实上，这些聊天机器人并不能真正理解用户的真实意图。想要理解用户交谈的目的与意图，要对用户的消息进行意图识别操作。

意图识别的方法也有多种：第一种，也即最简单的一种，是借助正则表达式匹配关键字信息获取意图。最简单粗暴的策略即，当句子中含有某一特定单词/某一类表达时，便可划定该类句子意图。举例而言，含有 “我好饿”的句子，正则表达式匹配后将其意图划分为“吃”。

第二种方法为借助监督学习的方法，将训练句子集在 SVM 分类器上训练。之后出现新的句子，使用分类器即可得出可能性最大的句子意图。

当然，也可按照最近邻近分类法，将未知意图的句子向量化后与其他各个意图群向量的中心进行比对，将其划分给中心距离最近的意图。

2.3 命名实体识别

命名实体识别是自然语言处理上基础但重要的任务，也是聊天机器人工作时必有的步骤。它指从文本中识别出命名性指称项，为后续关系抽取、依赖分析等任务做铺垫，包括识别人名、地名、机构、时间、货币名称等。

命名实体识别亦可以利用正则表达式匹配关键字、人名和时间等。但如果实体名字比较多，类型繁杂，用正则表达式不仅难于调配，更是繁琐，增加编程工作量。

同时也可以使用 spacy 工具进行预建的命名实体识别。spacy 分析后的 doc 对象的 ents 属性是一个列表，列表内各元素为文本中的命名实体，并已完成分类和标注。列表元素 ent 的原文本为 ent.text，实体标签分类为 ent.label_ 。

spacy 工具不仅可以识别与标注实体，还可以进行依赖分析，获得词之间的修饰关系。对于经 spacy 解析的文本对象 doc，doc[x].ancestors 是句子内标号为 x 的单词的语法树上的祖先，可用来分析词之间的依赖关系。

2.4 RASA——用于构建上下文聊天机器人的机器学习框架

除了 spaCy 外，另一搭建聊天机器人时超有用的框架工具是 Rasa。

Rasa 是一个开源的机器学习框架，用于构建上下文聊天助手/机器人。它包含两个大模块：1）Rasa NLU 和 2）Rasa Core。Rasa NLU 用于分析理解自然语义以及从文本中提取有效信息。Rasa Core 则是用于维持对话以及决定后续操作。

本项目开发本地聊天机器人主要使用 Rasa NLU 模块。使用前，将 pipeline 设为 "spacy_sklearn"，同时配置用于 nlu 训练的 json 训练数据文件：在其中列出一些聊天时可能出现的对话文本，并标记其意图和需要识别并提取的实体。注意，在新版本的 rasa 中，每一类意图都需要至少两条的训练文本。

在训练数据就绪后，便可以训练出 NLU 模型，可以用于识别、抽取训练数据所含类型的意图和实体。

2.5 数据访问与获取

本项目设计数据访问与获取的技术有两种：查询数据库和调用 API。

2.5.1 数据库系统

数据库系统是一套较为成熟的数据存储、数据查改和数据管理的系统方案，是软件系统开发时访问与获取数据的常用选择之一，尤其适合目标数据源已全部具有的情况。

使用时先建库、数据入库，然后在需要数据时，根据对应数据库操纵语句的语法，书写具有增删改查各种功能的语句，例如 SQLite 的 insert 和 select 语句可用与在库中添加和查找数据。然后再 python 中对写好的数据库操纵语句进行数据注入和封装，然后执行得到目标结果。本次使用关系数据库 SQLite。

通过数据库访问数据简单且方便，但缺点是只能访问目标数据库中有的数据，因此在数据库数据有限时，查询能力也十分有限。

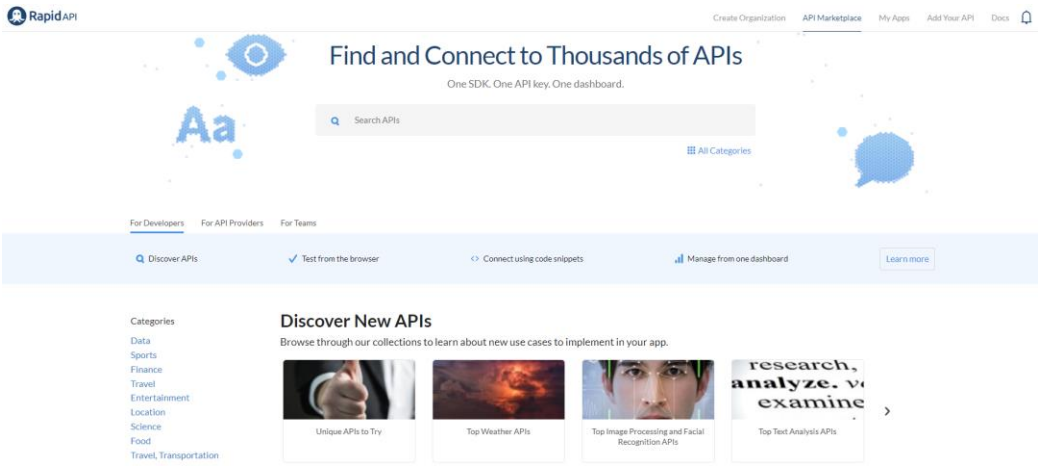
由于 api 获取数据在网络状态不佳时，效率较低，数据获取能力也有限，故本次项目结合了 api 和数据库两种方式，通过多次调用，已本地化了一部分数据，一定程度上增加了数据获取的效率。

2.5.2 应用程序接口：API

除了数据库系统外，API——应用程序接口，也是数据获取方式。

API 为暴露在互联网中的应用及程序接口，连接互联网的计算机在获得特定 API 得访问权限后，可通过 HTTP 请求的方式从该 API 获取数据。

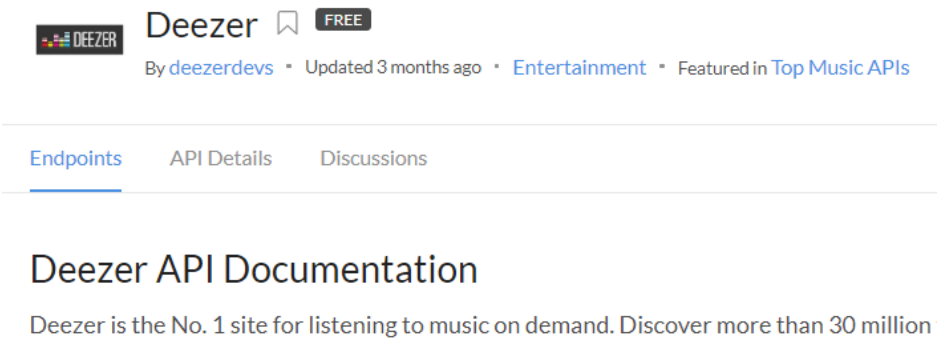
请求获取的数据类型很多种，API 得种类更是不可数。在老师提供我们参考的站点 rapidapi.com 上，存在大量优质开源的 API 资源可供使用。

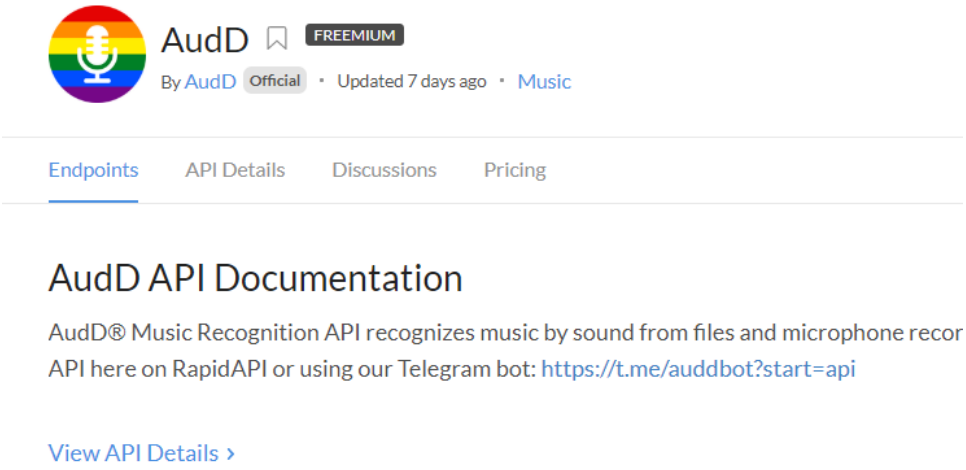


本次项目共使用了两类 api 接口：

其一为 Deezer 免费的接口，Deezer API 可根据歌曲名称搜索曲库得到接近的歌曲信息，也可根据 Deezer 平台歌曲唯一标识 id 对歌曲信息进行搜索，包括封皮、专辑信息、创作者等信息。同时，针对 Deezer 创作者的 id 也可搜索创作者详细信息。

另一 API 为 AudD 的听歌识曲接口，上传 audio 文件，即可获得歌曲信息、以及该歌曲在几大主流音乐市场（包括 Deezer、Spotify 等）的标志 id 和分享页。





本次开发结合了数据库技术和上述两 API 获取数据。

2.6 单轮多次问询与否定甄别

现实生活中，经常见到下面这种对话：

——“我想要一杯茶饮品”

——“我们这边有三种饮品：乌龙茶、龙井茶和毛尖茶，你想要哪个”

——“乌龙茶，谢谢”

在上面的对话中，顾客说了两句意图为“点茶饮品”的话，分两次补足了信息。也即，完成了一次单轮多次问询。

使用增量过滤器即可实现单轮多次问询。在单轮首次问询发生时，我们的后端已经可以通过意图识别和实体抽取，进而根据限定条件（也即实体的值）搜索和获取符合要求的数据。若实体抽取的结果不足以获取特定的数据，则需要进一步的限定条件。机器人便可以向用户发出提问，索要更多信息，或者根据先前限定范围内的多种选择，给用户选择范围。如此进行对话，在有限次数内，用户便可在问询中得到最终结果。

但是，按上述处理问询会碰到否定被忽视的问题：由于实体抽取时得到的限定条件均为肯定的条件约束，也即只要出现某一实体以及其值信息，便会在抽取得到的限定下进行数据查询搜索。

实际生活中，人们的话语中经常使用否定表达，用来表达范围和限定含义，如：“我想要一杯茶饮，哦对了，不要龙井茶”。按照上述的问询方式抽取的结果必定含有“茶”和“龙井”，进而问询结果为满足是 “乌龙茶”的茶饮，这显然是错误的。

所以，在聊天 bot 的开发中，需要对否定实体进行甄别。

本次开发甄别否定的方式较为简单，也即将原有的表达按照实体划分成多个句子“切片”，含有某一实体信息的切片中若含有语法程度上否定的表达（中文“不”，英文“no”、“not”、“n’t”等），即将该实体标记为否定限定，在之后访问数据时，限定取反。

例如上面的句子“我想要一杯茶饮，哦对了，不要龙井茶”会被分为“我想要一杯茶饮”和“，哦对了，不要龙井茶”。其中实体“茶饮”所在切片无否定，“龙井茶”有否定，访问数据时，便会排除分类为“龙井茶”的条目。

当然，由于人类的语言过于多变和复杂，否定表达的形式多样且多变，难以通过一种处理方式一劳永逸。上面的甄别否定的方法只适用于多数场景，并不能保证全部正确，例如在否定后置，或其他隐含否定词出现时会失效。

2.7 多轮多次问询

众所周知，在人与人聊天交谈的过程中，话语的意图并非一成不变的，而是随着沟通的过程不断变化的。因此，若机器人只能满足单轮多次问询的需要，是不能满足“智能”二字的要求的。聊天机器人不仅需要随着聊天意图的改变及时调整回复方式，还要体现逻辑，这时候就需要状态机来维护和变迁对话的状态。

状态机大家都不陌生，应用于聊天机器人时，其实只需将用户消息的意图准确识别，并把新消息的用于意图作为改变状态变迁的动作。因此，可以根据需要建立状态转换表，在 python 中便可以用二元组（状态，意图）对应（新状态，动作），建立状态变迁字典：

{ (状态 1, 意图 1): (状态 2, 动作 1),

(状态 2, 意图 2): (状态 3, 动作 2),

…}

可是，只用状态机用于维护轮询状态是不够的，因为事情不总是按照相同的情况发生。比如售卖聊天机器人的有些货物会售光。以可乐为例，若可乐已售光，若有用户要点可乐，该聊天机器人应提醒该用户可乐不足，并向客户推销库存充足的红茶。

这时候已不是简单固定的多轮多次问询了，而是有待定行动和等待状态的问询，同时，用户还可能会拒绝建议。

这种情况下，在状态变迁的动作部分应返回两个值，一个为动作 1：此动作立即执行，用于向客户提建议，另一个为待定动作（pending）：只有当客户答应了动作 1 的建议，才会执行待定动作，否则当前的待定动作清空。且初始时待定动作为空。

这样就能实现有等待状态转换和待定行动的多轮多次查询技术。

本次项目结合了状态机和待定行动两方法。但由于在音乐机器人中，涉及情况较多，数据情况也较为复杂，故进行状态转换时并未使用字典匹配的状态机，而是对状态和意图进行分类讨论，并结合搜索结果变迁状态和更改数据。

3 总结与收获

<removed>