

远程科研总结报告

李凌昊
2019.1.5

目录

1. 研究背景-----	3
2. 研究过程及成果-----	5
3. 学习感受与收获-----	9
4. 引用-----	11

1. 研究背景

随着计算机语言的和人工智能的飞速发展，智能聊天机器人已经成为当今如火如荼的一项科研项目。智能聊天机器人是一项综合计算机、电子、人工智能、自动控制、系统科学等等的综合性研究项目。结合了许多先进的计算机理念以及硬件技术，智能聊天机器人能够极大的简化处理步骤，提高生产效率。

何为聊天机器人？维基百科给出的广泛定义是：聊天机器人（chatbot）是经由对话或者文字进行交谈的电脑程序，能够模拟人类对话，通过图灵测试。聊天机器人可用于实用的目的，如客户服务或者资讯获取，有些聊天机器人会搭载自然语言处理系统，但只会截取输入关键字，再从数据库总寻找最合适的应答句。目前，聊天机器人是虚拟助理的一部分，可以与许多组织的应用程式，网站以及即时消息平台链接。非主力应用程式包括玉律目的的聊天室，研究和特定产品促销，社交机器人等等。

目前的聊天机器人通常整合与虚拟助理当中，使其能够与客户进行范畴内的聊天以及数据交流。目前聊天机器人广泛运用于即时通讯平台，例如脸书 Messenger, WeChat, LINE 和 Kik, 以娱乐，零售行销，以及客服为目的。此外，即时通讯平台提供易于整合的 webhook, 使得第三方开发商易于可通用于不同通讯平台之聊天机器人。[1][2] 这些软体机器人以客服的身份出现或是成为团体聊天的一员。有些即时通讯的机器人可以连接外部资料库，提供使用者新闻，气象，导航，电影放映时间，股价等资讯。[13]达美乐，必胜客，迪士尼，Nerdify, 雅玛多 Line, 全食超市都已推出各自的聊天机器人，以便与终端消费者增进交流，推销公司的产品与服务，并且让消费者订货更加方便。[3] 2016 年，观光业的一些旅行社和航空公司透过

Messenger 推出了聊天机器人的服务，墨西哥航空利用人工智能售票，回答问题，墨航和荷兰皇家航空并且提供航班资讯，处理乘报道，发出行动登机证，推荐旅馆，餐厅，目的地行程。中国的旅行社在此之前已用用微信提供这些服务。路易莎咖啡运用 EILIS 智慧互动助理聊天机器人技术建立虚拟黑卡让营收倍增。

在教育方面，智能聊天机器人的效率性也得到了充分的体现。例如 Nerdify 开发的 Nerdy Bot，针对大中小学生面对的问题，让学习更简单又有效率。

[1] [2]该软体利用脸书 Messenger 即时回答学生作业相关的问题以便加速学习。加大尔台湾分校图书馆的聊天机器人 ANTswers，2014 年开始试用，被认为非常成功。[3]

正因为智能机器人在商业科研上体现出的重要价值，突显出我们对这门学科学习与研究的特殊价值，在此科研中，我们将利用人工智能，自然语言处理理解的方方面面，来学习运用于具体的项目研究中。

2. 研究过程及成果

项目要求：

1. 采用问询股票信息为应用背景的金融聊天机器人作为课程大作业。需要自己找到相应 API 来实现查询（比如携程网或者饿了么的 API 来进行查询）
2. 代码里尽量多的采用上述九项实现技术要求至少能够实现三种意图的查询（比如问询某几种股票的股价信息、成交量信息、市值等），每种意图需要至少两轮查询从而得到查询结果；
3. 本次训练利用英文完成询问和回答即可，需要自行按照 Rasa NLU 中的训练数据为例子，自行构建训练数据并训练处模型来使用；
4. 可以通过以下 iexfinance 的 API 的示例来进行查询实时股票价格，交易量，开盘价等各种信息。

解决办法：

- 1.模拟对话情景，训练 Rasa NLU（得到响应语句模型）
- 2.多轮多次查询功能完善（实现三种意图的查询，比如问询某几种股票的股价信息、成交量信息、市值等）两轮查询，+ IEX API
- 3.把写好的文件 wxpy 上面去与微信进行整合

代码截图：

```
pip install rasa_nlu[tensorflow]
```

控制栏：

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

代码：

```
from rasa_nlu.model import Interpreter
import json
```

```
interpreter = Interpreter.load("./models/current/nlu")
message = "let's see some apple price"
data = interpreter.parse(message)
print(data)
```

```
def send_message(state, message):
    print("USER : {}".format(message))
    new_state, response = respond(state, message)
    print("BOT : {}".format(response))
    return new_state
```

```
def respond(state, message):
    (new_state, response) = policy_rules[(state, interpret(message))]
    return new_state, response
```

```
{'intent': {'name': 'stock_search', 'confidence': 0.95255047082901},
 'entities': [{'start': 15, 'end': 20, 'value': 'apple', 'entity': 'company', 'confidence':
0.8437595026254843, 'extractor': 'ner_crfs'}, {'start': 21, 'end': 26, 'value': 'price', 'entity': 'price',
'confidence': 0.8920190645015604, 'extractor': 'ner_crfs'}],
'intent_ranking': [{'name': 'stock_search', 'confidence': 0.95255047082901},
{'name': 'greet', 'confidence': 0.15953731536865234},
{'name': 'affirm', 'confidence': 0.05648072808980942},
{'name': 'thankyou', 'confidence': 0.0}],
'text': "let's see some apple price"}
```

```

def interpret(message):

    if 'data.get('intent').get('name')' == 'stock_search':
        return 'stock_search'
    if 'kenyan' in msg or 'columbian' in msg:
        return 'specify_coffee'
    if 'what' in msg:
        return 'ask_explanation'
    return 'none'

# Define the states
GREETING = 0
INIT = 1
STOCK_SEARCH = 2
CHOOSE_STOCK_PRICE = 3
CHOOSE_TRADING_VOLUME = 4
THANK_YOU = 5

# Define the policy rules dictionary
policy_rules = {
    (INIT, "GREETING"): (INIT, "Hello. I'm a bot to help you to check stock information for specific
company, you can ask me for stock price or trading volume."),
    (INIT, "STOCK_SEARCH"): (INIT, "OK! You can ask me for stock price or trading volume "),
    (STOCK_SEARCH, "CHOOSE_STOCK_PRICE"): (STOCK_SEARCH, "do you want more
information?"),
    (CHOOSE_STOCK_PRICE, ""): (ORDERED, "perfect, the beans are on their way!"),
    (CHOOSE_COFFEE, "ask_explanation"): (CHOOSE_COFFEE, "We have two kinds of coffee
beans - the Kenyan ones make a slightly sweeter coffee, and cost $6. The Brazilian beans
make a nutty coffee and cost $5.")
}

# Define send_messages()
def send_messages(messages):
    state = INIT
    for msg in messages:
        state = send_message(state, msg)

# Send the messages
send_messages([
    "I want to know about apple stock",
    "I would like to see the stock price of apple",
    "Yes",

```

```

    "I also want to see the market value of apple",
    "Thank you, you helped me a lot"
})

```

```

USER: Hello
BOT: Hello. I'm a bot to help you to check stock information for specific company, you can ask
me for stock price or trading volume GREATING_STATE
USER: I want to know about apple stock IDENTIFY_COMPANY_NAME_STATE
BOT: Okay, What kind of information do you want to see. You can ask me stock price or market
value CHOOSE_STOCK_INFO
USER: I would like to see the stock price of apple
BOT: The stock price for APPLE is xxx.xx. STOCK_PRICE_STATE
BOT: Do you want to know other informations? ASK_MORE_INFO
USER: Yes
BOT: Okay, What kind of information do you want to see. You can ask me stock price or market
value CHOOSE_STOCK_INFO
USER: I also want to see the market value of apple
BOT: The market value for APPLE is xxx.xx. MARKET_VALUE_STATE
BOT: Do you want to know other informations? ASK_MORE_INFO
USER: No, Thank you, you helped me a lot
BOT: Thank you for using stock check bot GREATING_STATE

```

```

def interpret(message):
    if data.get('intent').get('name') == 'greet':
        return 'greet'
    if data.get('intent').get('name') == 'stock_search':
        if data.get("entities").get("entity") == "price":
            return 'stock_price'
        if data.get("entities").get('entity') == "volume"
            return 'volume'
        if data.get("entities").get('entity') == ""
            return "none"

INIT = 0
CHOOSE_STOCK_PRICE = 2
CHOOSE_TRADING_VOLUME = 3
THANK_YOU = 4

# Define the policy rules dictionary
policy_rules = {

    (INIT, "greet"): (INIT, "Hello. I'm a bot to help you to check stock information for specific
company, you can ask me for stock price or trading volume"),
    (INIT, "stock_price"): (INIT, ""),
    (CHOOSE_COFFEE, "specify_coffee"): (ORDERED, "perfect, the beans are on their way!"),
    (CHOOSE_COFFEE, "ask_explanation"): (CHOOSE_COFFEE, "We have two kinds of coffee
beans - the Kenyan ones make a slightly sweeter coffee, and cost $6. The Brazilian beans
make a nutty coffee and cost $5.")
}

```

总结：

通过代码的运行，我们小组能够成功的使用微信小程序来进行程序的运行并且获得我们想要的股票价格。在将来我们的改进方向是进行多语言，多种逻辑的问答升级，来提高用户的体验。

3. 学习感受与收获

在报名参加只能聊天机器人之前、深度学习、智能神经网络等等一切对我而言都是一个朦胧而又遥远的概念。作为一个工程学院的学生，我只知道随着科技的迅速发展，只能聊天机器人的应用以及普及会越来越广，其所蕴含的不仅仅是巨大的商业价值，更拥有无法比拟的科研价值。

短短一个月紧张而充实的学习以及参与让我对于聊天机器人有了全面的认识，每一节课堂以及课后作业都充满了全新的知识内容以及更高层次的挑战。充实感慢慢替代了刚开始的紧张以及焦虑。刚开始一周我了解了人工神经网络，循环神经网络、序列分析算法、序列生成算法和自然语言内容提取，并以 Tensorflow 为主要编程框架熟悉其相关模块功能和使用。再紧接着深入学习聊天机器人算法的三驾马车:自然语言序列 到序列算法、编码解码结构、注意力模型，在垂直领域的公开数据集上训练出具有一定能力的自我应答系统。在后续的设计股票查询聊天机器人当中，我们组边钻研边学习了侧重于智能应答系统的专属记忆网络，采用机器学习模型对记忆网络进行优化并能构建出具备长期记忆能力的深度记忆模型，实现应答准确性和应答速度的性能优化。在老师的帮助以及悉心指导下，我们初步完成了微信小程序中的智能股票查询聊天机器人，锻炼了我独立学习和合作研究的能力，提高了我对于计算机语言的思维能力。这次科研的经历也让我明白了，遇到困难和问题不要轻易放弃，要想办法克服困难、解决问题。要善于思考、多动手动脑。感谢老师在这一个月时间里的耐心教导，让我学到了更多的关于聊天机器人知识，也体会到了聊天机器人在未来对社会发展的重要意

义。我将会运用我在此次科研中学到的知识运用到实践中去，把它作为工具来应对科研商业发展的需要。

4. 引用

[1]: What is a chatbot?. techtarget.com. [30 January 2017].

[2]: <http://gizmodo.com/google-assistant-is-a-mega-chatbot-that-wants-to-be-abs-1777351140>

[3]: Kane, D.A. The Role of Chatbots in Teaching and Learning. (编) Rice, Scott; Gregor, M.N. E-Learning and the Academic Library: Essays on Innovative Initiatives. McFarland & Co. 2016: 131–147. ISBN 978-1-476-62441-9.