微信公众号智能聊天机器人实现（基于南开信息构建语料库）

李伟

1711350 计算机科学与技术一班

更新：*December 28, 2019*

摘　　要

现在，人工智能日益繁荣，相关的技术实践和产品也很多，基于信息检索相关的技术，结合爬虫，文本分析，自然语言处理等相关技术集成形成智能聊天机器人，聊天机器人虽然本质不是特别复杂和核心的应用，但是作为科研科应用的交互点以及人机交互的重要方式，其意义不言而喻，本次实验基于网络爬虫技术，信息检索基本思想，基础的自然语言处理知识，网络应用部署，服务器配置部署，微信平台应用外接等相关理论知识和技术手段，完全自主的实现了微信公众号后台聊天机器人的搭建，实现用户与智能机器人便捷有效的交流。

关键词： 网页爬取，人工智能，聊天机器人，微信公众号后台开发，网络应用服务部署，chatterbot

1　实验要求与问题阐述

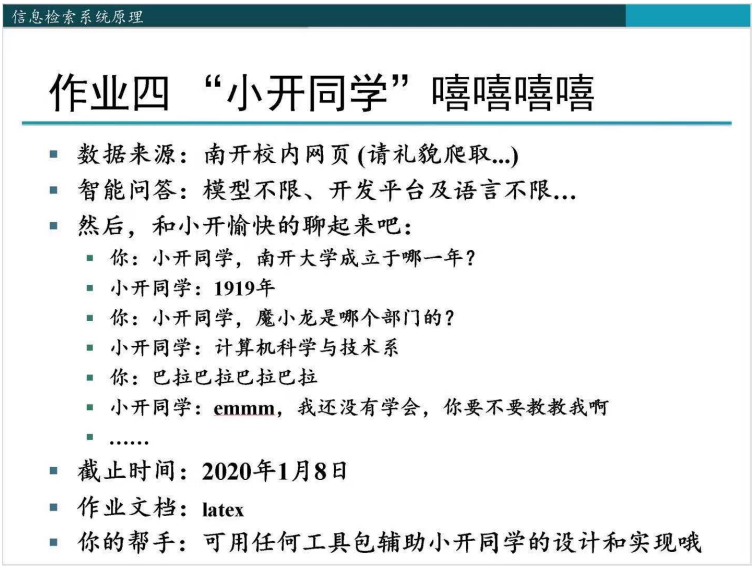

图 1: 作业要求

如图 1所示，本次作业要求基于南开大学的相关信息，结合信息检索相关知识实现一个语音聊天机器人，能够和用户进行交流。

- 网页抓取：需要抓取一定的数据进行语料库构建
- 语料库构建：要求对网页及其锚文本，以及相关的信息域进行解构，生成一些语料库
- 模型构建：可以采用任何模型构建聊天机器人
- 聊天服务：可以自行实现一些聊天平台服务
- 实现文档：要求编写实现文档

针对此次作业要求，主要可以分为网页爬取，预料库构建，模型建立，聊天服务实现，四个主要工作模块，之后的实现也基本按照这个流程和分类实现。

2 实验环境

- 开发语言：python
- 开发平台：pycharm
- 主要工具包使用：numpy，scrapy，web,chatterbot, NLTK……
- 模型构建基于：chatterbot 聊天机器人开源包
- 开发系统环境：window 10 专业版
- 服务器系统环境：Ubuntu 18.04
- 服务器：华为云服务器
- 运行平台：微信公众号后台

3 实现思路和步骤

首先，本次实验主要可以分为网页爬取，预料库构建，模型建立，聊天服务实现，四个主要工作模块，以下主要从这四个方面介绍本次实验的实现思路和具体的实现步骤。

3.1 数据爬取

本次数据爬取采用第三次作业的爬取代码和相关数据，主要构建关于南开大学教师相关的语料库，爬取的数据的截图如**图 2**所示：

图 2: 数据爬取截图

3.2 语料库构建

本次的训练语料库包括自定义语料库以及开源语料库，开源语料库使用的是 chatterbot 的中文语料库，质量较高，自定义语料库基于第三次作业爬取的教师数据，构建语料库的代码如下所示：

```
1
2  from data import get_teacher_info,get_content
3
4  # 预料库存储位置
5  corpus = "./corpus"
6
7  # 生成南开大学教师基本学院相关信息问答语料库
8  def create_teacher_info_corpus():
9  info = get_teacher_info()
10 print(info)
11 teacher = open("./corpus/teacher.yml",'w',encoding='utf-8')
12 teacher.write("categories:\n")
13 teacher.write("- teacher\n")
14 teacher.write("conversations:\n")
15 xueyuan = open("./corpus/xueyuan.yml",'w',encoding='utf-8')
16 xueyuan.write("categories:\n")
17 xueyuan.write("- xueyuan\n")
18 xueyuan.write("conversations:\n")
19 for key in info.keys() :
20 teacher.write("- - "+info[ key]["name"] +"\n  - "+info[key]["name"]+"老
     师是"+info[key]["xueyuan"]+"的\n")
21 teacher.write("- - " + info[key]["name"] + "是哪个学院的\n  - " + info[
     key]["name"] + "老师是" + info[key]["xueyuan"] + "的\n")
22 teacher.write("- - " + info[key]["name"] + "在哪个学院\n  - " + info[
     key]["name"] + "老师是" + info[key]["xueyuan"] + "的\n")
23 teacher.write(
24 "- - 我想去" + info[key]["name"] + "老师主页看一下\n  - 这是" + info[
     key]["name"] + "的主页链接" + info[key]["url"] + "\n")
25 teacher.write(
26 "- - " + info[key]["name"] + "老师主页\n  - 这是" + info[key]["name"] +
      "的主页链接" + info[key]["url"] + "\n")
27 teacher.write(
```

```
28 "- - " + info[key]["name"] + "主页链接\n  - 这是" + info[key]["name"] +
       "的主页链接" + info[key]["url"] + "\n")
29 teacher.write(
30 "- - 你知道" + info[key]["name"] + "老师吗\n  - 当然知道, " + info[key
   ]["name"] + "是"+info[key]["xueyuan"]+"的, 这是她的主页链接" + info[
   key]["url"] + ", 想知道更多的信息可以点击一下\n")
31
32 xueyuan.write("- - 我想去" + info[key]["xueyuan"] + "主页看一下\n  - 这
   是" + info[key]["xueyuan"] + "的主页链接" + info[key]["parentUrl"] )
33 xueyuan.write(
34 "- - " + info[key]["xueyuan"] + "主页\n  - 这是" + info[key]["xueyuan"]
       + "的主页链接" + info[key]["parentUrl"] )
35 xueyuan.write(
36 "- - " + info[key]["xueyuan"] + "\n  - 这是" + info[key]["xueyuan"] + "
   的主页链接" + info[key]["parentUrl"] )
37 xueyuan.write(
38 "- - 你知道" + info[key]["xueyuan"] + "吗\n  - 我当然知道, 这是" + info
   [key]["xueyuan"] + "的主页链接" + info[key]["parentUrl"] )
39 teacher.close()
40 xueyuan.close()
41 return 0
42
43 # 生成计算机学院教师信息对话语料库
44 def create_cc_info():
45 info = get_teacher_info()
46 cc = open("./corpus/cc.yml", 'w', encoding='utf-8')
47 cc.write("categories:\n")
48 cc.write("- cc\n")
49 cc.write("conversations:\n")
50 titles = ['性别','所属部门''行政职务','职称','学历','所学专业','办公电
   话','电子邮件','研究方向']
51 for key in info.keys() :
52 if info[key]['xueyuan'] != "南开大学计算机学院":
53 continue
54 t_f_path = get_content(info[key]['xueyuan'],info[key]['name'])
```

```
t_f = open(t_f_path,'r',encoding='utf-8')
for line in t_f :
tt = line.split(": ")
print(tt)
if tt[0] in titles:
cc.write("- - " + info[key]["name"]+"的"+tt[0] + "是什么\n  - " + info[key]["name"] + "老师的"+tt[0]+ "是"+
tt[1].strip('\n')+ "\n")
t_f.close()
cc.close()
return 0

if __name__ == "__main__":
create_teacher_info_corpus()
create_cc_info()
```

从代码注释中可以看到，自定义的语料库主要是教师的个人信息和学院主页链接信息，生成的语料库问题和回答比较僵硬，所以训练的效果不是很好，但是基本可以使用。上述代码生成的语料库样例截图如**图 3**所示：

图 3: 自定义预料可截图样例

3.3 模型建立

模型建立依靠 chatterbot 开源工具包，实现对开源语料库和自定义语料库的训练工作，实现聊天机器人的功能，首先构建聊天机器人对象，代码如下所示：

```
from chatterbot import ChatBot,comparisons,response_selection,languages,filters
# 初始化聊天机器人设置
myChatBot = ChatBot(
```

```
"ChatterJack",
storage_adapter='chatterbot.storage.SQLStorageAdapter',
logic_adapters=[
{
'import_path': 'chatterbot.logic.BestMatch',
'default_response': '我没明白你的意思诶，不过你可以问我南开教师有关的问题，我对计算机学院的老师很熟悉哦！我还会说笑话唱歌呢！',
'maximum_similarity_threshold': 0.90
},
{
'import_path': 'chatterbot.logic.SpecificResponseAdapter',
'input_text': '在吗？',
'output_text': '我一直都在你的心里^_^',
}
],
statement_comparison_function=comparisons.levenshtein_distance,
# statement_comparison_function=comparisons.SynsetDistance,
preprocessors=[
'chatterbot.preprocessors.clean_whitespace',
'chatterbot.preprocessors.unescape_html',
],
filters=[filters.get_recent_repeated_responses],            # 去除最近重复的回应，避免机器人重复说一样的话
response_selection_method = response_selection.get_first_response,
#response_selection_method = response_selection.get_random_response,
# input_adapter='chatterbot.input.VariableInputTypeAdapter',
# output_adapter='chatterbot.output.OutputAdapter',
read_only=True,    # 避免在学习每一次对话输入
)
```

上述的设置均是根据 chatterbot 官方文档的说明添加的，能够实现不同的聊天功能，实现不同的处理。比如特定问题的特定回答，默认回答等等。

之后进行训练，在本实验中我才用了两种训练方式：语料库文本 yml 格式文件训练，对话列表训练，训练代码如下所示：

```

# train.py
```

```
from chatbot import myChatBot
from chatterbot.trainers import ChatterBotCorpusTrainer
# 设置训练器
trainer = ChatterBotCorpusTrainer(myChatBot)
# 使用现有的中文语料库训练它，具备初始的问答能力
#trainer.train("chatterbot.corpus.chinese")  # 中文语料库
# 使用自定义的的中文语料库训练它
trainer.train("./corpus/")  # 中文语料库



# retrain.py
from chatbot import myChatBot
from chatterbot.trainers import ListTrainer
from data import get_text
#设置训练器，采用句子序列的方式训练
trainer = ListTrainer(myChatBot)

trainer.train([
"你好",
"你好，我是你的专属机器人Jack！",
"你会做什么？",
"上知天文，下知地理，无所不知无所不晓！",
"这么厉害啊！那你知道南开大学吗？",
"当然知道，南开大学是主人读的大学啊，我最了解她了！",
"南开大学",
"南开大学是一所著名的双一流大学",
"南开大学是周恩来的母校",
"谁是最帅的人？",
"当然是我的主人您啊！",
"谁最帅？",
"当然是李伟啊！我的主人天下第一帅！",
])

trainer.train(get_text())
```

3.4 聊天服务

本次聊天服务才用微信公众号的开发者接口实现，将聊天机器人接入微信后台控制接口，实现自定义回复用户的消息，从而达到和用户便捷交流的目的。聊天的对话页面样例截图如图 4所示：

图 4: 聊天界面截图

响应微信消息的 web 服务基于开源 web.py 实现，基本可以参考微信开发者文档中进行编写。部分代码截图如下所示：

```
# -*- coding: utf-8 -*-
# filename: main.py
import web
from handle import Handle

urls = (
'/wx', 'Handle',
)

if __name__ == '__main__':
app = web.application(urls, globals())
app.run()
```

此代码来自于 main.py 文件，运行该文件启动 web 服务响应。

3.5 服务器一键部署

为了更方便将 web 服务搭建到服务器上，我编写了部署文件 start.sh，在服务器终端运行该部署文件，则课快速进入虚拟环境部署运行 web 服务，不必多次输入相似的命令，方便部署构建，服务器构建 Python 虚拟环境，安装 chatterbot，re，chatterbot-corpus 等工具包和语料库，之后即可后台运行该服务，就可以在微信后台和聊天机器人进行交互了。部署文件代码如下：

```
#!/bin/bash     ---指定解释器
echo "Begin deploy chatbot"     --- 输出提示信息
source venv/bin/activate      --- 进入虚拟Python环境
python train.py           --- 开始进行文件格式语料库训练
python retrain.py         --- 开始进行对话列表训练
nohup python main.py 80 &     --- 后台运行进程
echo "finish deploy"        --- 输出结束提示信息
```

4 结果展示

这个结果展示主要就就是聊天记录了，接下来就贴几张图微信后台聊天截图吧

图 5: 微信后台聊天截图 1

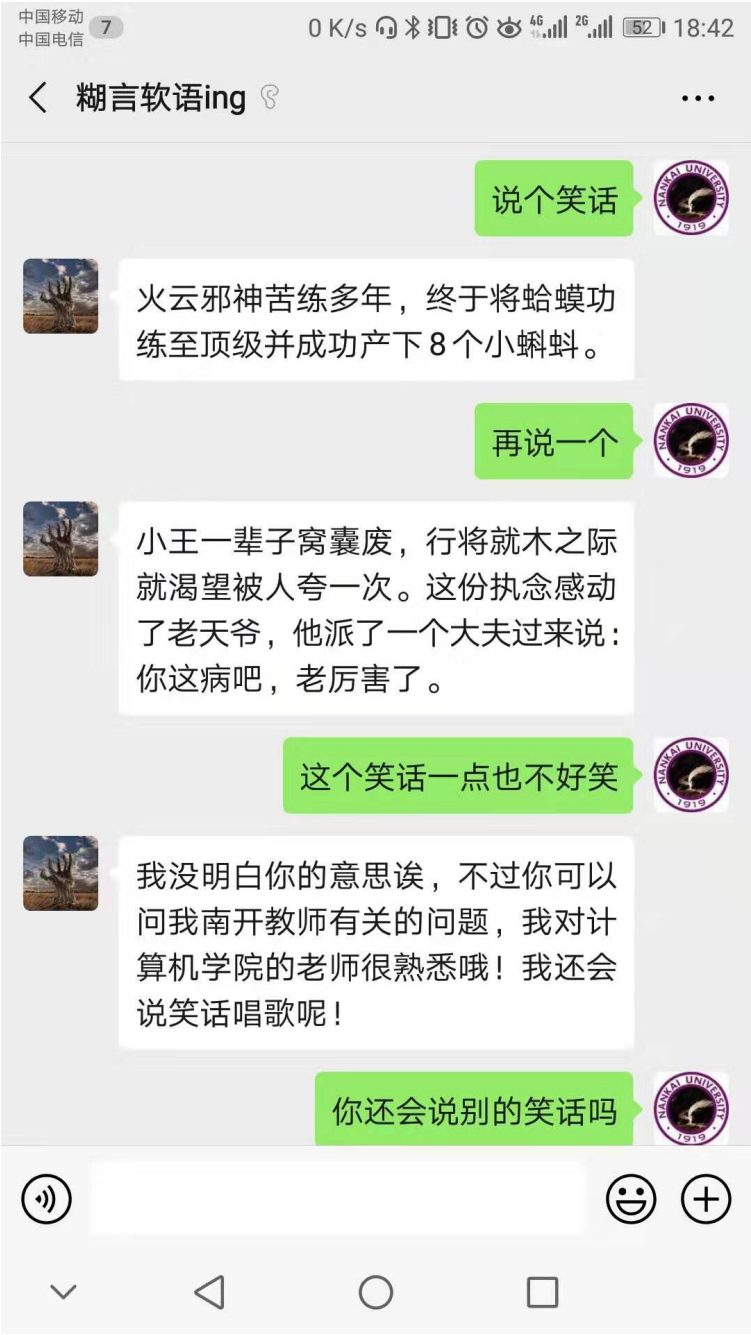

图 6: 微信后台聊天截图 2

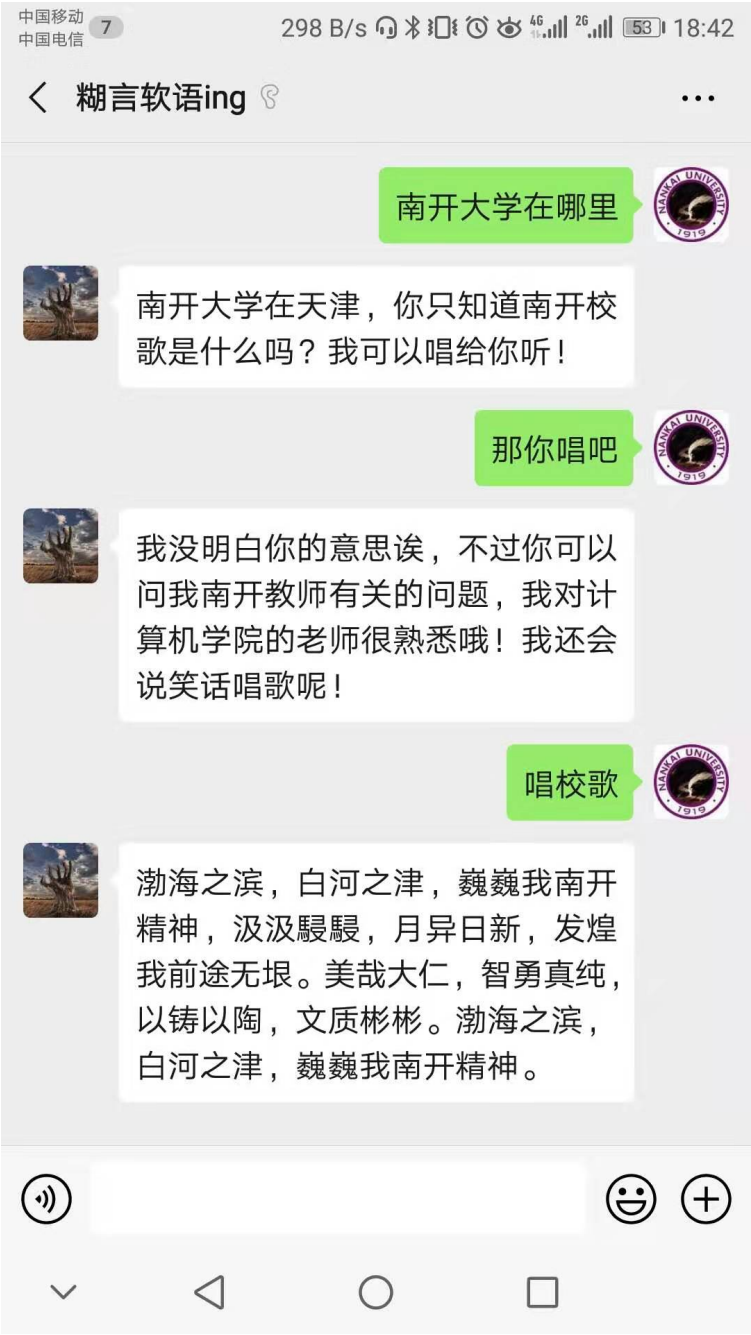

图 7: 微信后台聊天截图 3

5 用户体验

图 8: 部分体验用户关注公众号截图

- 李双庆：还行，能回答老师相关的一些信息
- 李时：问着问着就不会聊了，很多超出他能理解的问题就不能返回正确的答案
- 李力挺：可以回答预设的问题提问方式，但灵活性不够
- 周辰霏：可以回答教师相关的基本提问，也可以抖一下机灵，但很多提问不能理解，可能是语料库比较小。

6 总结与思考

通过这次的聊天机器人实现，我学会了很多新的技能，第一次搭建服务器服务，和使用微信的开发者接口，虽然用了一些时间，但是还是有很多收获的，虽然最后自己构建的机器人和图灵机器人接口的回答相比还是有很多问题，但是至少是自己从头到尾，从模型构建到平台接入全流程实现的，成就感还是很大的，而且确实也感觉很有意思很好玩。不过总算写完了，自己也算比较满意，就是 chatterbot 的模型不是很精确，不能够很好的进行学习，回答的准确性不好，我想图灵机器人接口应该是使用了机器学习，而且语料库也更加的丰富，所以自然比不了，等之后有时间可以再丰富一下语料库，应该表现会更好一点，因为这次期末快到了，没有太多时间去做更多更高质量的语料库了，所以就先这样吧。

7 latex 源码截图

```
%!TEX program = xelatex
% 完整编译方法 1 pdflatex -> bibtex -> pdflatex -> pdflatex
% 完整编译方法 2: xelatex -> bibtex -> xelatex -> xelatex
\documentclass[lang=cn,11pt]{elegantpaper}
\usepackage{listings}
\usepackage{fontspec}
\setmonofont{Consolas}

\title{微信公众号智能聊天机器人实现 基于南开信息构建语料库)}
\author{\href{https://github.com/Jack-Lio}{李伟}}

\institute{1711350   计算机科学与技术一班}

% 不需要版本信息，直接注释即可
%\version{0.07}
% 不需要时间信息的话，需要把 \today 删除。
\date{\today}


% 如果想修改参考文献样式，请把这行注释掉
\usepackage[authoryear]{gbt7714}  % 国标

\begin{document}

\maketitle

\begin{abstract}
\noindent
现在，人工智能日益繁荣，相关的技术实践和产品也很多，基于信息检索相关的技术，结合爬虫，文本分析，自然语言处
理等相关技术集成形成智能聊天机器人，聊天机器人虽然本质不是特别复杂和核心的应用，但是作为科研科应用的交互的
以及人机交互的重要方式，其意义不言而喻，本次实验基于网络爬虫技术，信息检索基本思想，基础的自然语言处理知识
，网络应用部署，服务器配置部署，微信平台应用外接等相关理论知识和技术手段，完全自主的实现了微信公众号后台聊
天机器人的搭建，实现用户与智能机器人便捷有效的交流。
\keywords{网页爬取，人工智能，聊天机器人，微信公众号后台开发，网络应用服务部署，chatterbot}
\end{abstract}
```

图 9: 源码 1

```
    \item 模型构建：可以采用任何模型构建聊天机器人
    \item 聊天服务：可以自行实现一些聊天平台服务
    \item 实现文档：要求编写实现文档
\end{itemize}



针对此次作业要求，主要可以分为网页爬取，预料库构建，模型建立，聊天服务实现，四个主要工作模块，之后的实现也
基本按照这个流程和分类实现。

\section{实验环境}

\begin{itemize}
    \item 开发语言：python
    \item 开发平台：pycharm
    \item 主要工具包使用：numpy，scrapy，web,chatterbot，NLTK……
    \item 模型构建基于：chatterbot聊天机器人开源包
    \item 开发系统环境：window 10专业版
    \item 服务器系统环境：Ubuntu 18.04
    \item 服务器：华为云服务器
    \item 运行平台：微信公众号后台
\end{itemize}

\section{实现思路和步骤}
首先，本次实验主要可以分为网页爬取，预料库构建，模型建立，聊天服务实现，四个主要工作模块，以下主要从这四个
方面介绍本次实验的实现思路和具体的实现步骤。

\subsection{数据爬取}
本次数据爬取采用第三次作业的爬取代码和相关数据，主要构建关于南开大学教师相关的语料库，爬取的数据的截图如\f
igref{fig:1}所示：


\begin{figure}[htbp]
    \centering
    \includegraphics[width=0.6\textwidth]{43}
```

图 10: 源码 2