



**本科毕业设计（论文）**

北华大学综合教务系统

学 号：19

学生姓名：孙晓聪

指导教师：贺薪宇

所在学院：计算机科学技术学院

所学专业：软件工程专业

2018年6月

# 摘 要

本文论述了基于机器学习的微信聊天机器人的设计与实现，主要从概述，方案论证，需求分析等几个方面循序渐进地论述。近几年机器学习的热潮不断高涨，基于机器学习的应用不断出现，例如Google Assistance，Apple Siri，这些应用的出现进一步证实了机器学习具备了走出实验室，进入生产环境的条件。基于机器学习的微信聊天机器人主要是做一个可以代替微信用户被动与好友聊天的聊天机器人。主要设计思路：首先用户将微信账号与该系统绑定，用户上传以前聊天内容或者其他内容以供训练聊天机器人，系统被动接受用户微信好友聊天内容，最后通过RNN产生被动回复内容并发送给微信好友。该项目主要分为两个子项目：EVE（前端Web APP），MO（后端服务）。主要技术栈：前端React 搭配 Redux 构建单页面Web APP，后端 Python Flask 搭配 Tensorflow 和 itchat 实现该项目的主要业务服务。该项目遵循Agile 开发的思想，采用迭代，循序渐进和CI/CD的软件开发方式，旨在提供高质量，响应变化等特性的软件。

**关键词：**RNN，Tensorflow，React，Flask，Agile，CI/CD

# Abstract

This article discusses the design and implementation of the Wechat chat robot based on machine learning. It mainly discusses gradually from several aspects such as overview, program demonstration and demand analysis. In recent years, the upsurge of machine learning has continued to rise, and applications based on machine learning continue to emerge. Such as Google Assistance and Apple Siri, the emergence of these applications further confirms that machine learning has the conditions to get out of the laboratory and into the production environment. The machine learning-based WeChat chat robot is mainly a chat robot that can replace the WeChat users and chat with their friends passively. The main design idea: First, the user binds the WeChat account with the system. The user uploads the previous chat content or other content for training the chat robot. The system passively accepts the user WeChat friend chat content, and finally generates a passive reply content through the RNN and sends it to the WeChat friend. . The project is divided into two sub-projects: EVE (front-end Web APP) and MO (back-end service). The main technology stack: front-end React with Redux builds a single-page Web APP, and back-end Python Flask with Tensorflow and itchat to achieve the main business services of the project. The project follows the ideas developed by Agile and adopts iterative, step-by-step and CI/CD software development methods to provide high-quality, responsive changes and other features.

**Key words：**RNN，Tensorflow，React，Flask，Agile，CI/CD

**目 录**

[**摘 要 I**](#_Toc485040323)

[**Abstract II**](#_Toc485040324)

[**第1章 概述 1**](#_Toc485040325)

[1.1 课题的来源，目的，意义 1](#_Toc485040326)

[1.1.1 课题的来源 1](#_Toc485040327)

[1.1.2 选择课题的目的 1](#_Toc485040328)

[1.1.3 选择课题的意义 1](#_Toc485040329)

[1.2 国内外发展情况 2](#_Toc485040330)

[**第2章 方案论证 3**](#_Toc485040331)

[2.1 方案的提出 3](#_Toc485040332)

[2.2 可行性分析 3](#_Toc485040333)

[2.2.1 经济可行性 3](#_Toc485040334)

[2.2.2 技术可行性 3](#_Toc485040335)

[2.2.3 操作运行可行性 3](#_Toc485040336)

[2.2.4 法律可行性 4](#_Toc485040337)

[2.3 方案的选择 4](#_Toc485040338)

[**第3章 需求分析 5**](#_Toc485040339)

[3.1 功能需求分析 5](#_Toc485040340)

[3.2 性能需求分析 5](#_Toc485040341)

[3.3 系统数据流图和数据字典 6](#_Toc485040342)

[3.4 系统E-R图 8](#_Toc485040343)

[3.5 系统用例图 11](#_Toc485040344)

[**第4章 设计论述 13**](#_Toc485040345)

[4.1 概要设计 13](#_Toc485040346)

[4.2 数据库设计 13](#_Toc485040347)

[4.3 详细设计 16](#_Toc485040348)

[4.3.1登录模块 16](#_Toc485040349)

[4.3.2管理员模块 21](#_Toc485040350)

[4.3.3卖家模块 26](#_Toc485040351)

[4.3.4买家模块 28](#_Toc485040352)

[4.3.5物流模块 29](#_Toc485040353)

[4.3.6定时任务模块 31](#_Toc485040354)

[**第5章 软件测试与结果分析 34**](#_Toc485040355)

[5.1 测试的目标 34](#_Toc485040356)

[5.2 模块测试 35](#_Toc485040357)

[5.3 测试总结 36](#_Toc485040358)

[**结 论 37**](#_Toc485040359)

[**致 谢 38**](#_Toc485040360)

[**参考文献 39**](#_Toc485040361)

[**附 录 40**](#_Toc485040362)

# 第1章 概述

## 1.1 课题的来源、目的、意义

### 1.1.1 课题的来源

之前在某公司实习的过程中，公司沟通事项大多通过微信完成，并且沟通的内容大多是模板内容。这样的沟通不仅占用大量的时间而且没有太大实际意义。联想到当下比较热门的Google Assistance 等聊天助理，为什么微信没有一个类似的聊天助理来帮助用户从大量的琐事中解脱出来？刚好自己曾今看过一个可以将微信聊天操作API化的类库，这个类库让基于微信的程序聊天可以实现。带着这样的思绪与导师沟通后，加入了用机器学习相关技术处理用户聊天来避免陷入死板聊天，最后课题定为基于机器学习的微信聊天机器人的设计与实现。

### 1.1.2 选择课题的目的

经历过三年的基础知识的学习以及大四上学期校外实习的历练，迫切想通过一个完整的项目来整合自己所学的知识，同时本科毕业设计是本科阶段毕业前夕的关键一步。

微信作为中国最大的即时聊天工具之一，不论是用户还是社区都有举足轻重的地位。当然课题选为微信聊天机器人并不是想作为一个商业项目，而是带有实践性的一个社会实践项目。该项目需要的技术涉及前端，后端和机器学习等相关内容，前端和后端的技术在之前的学习中有过一定的基础知识积累，这次的毕业设计正好将两个技术栈有机的结合起来，加深对其的理解程度。机器学习是自己很渴望学习的一门新兴技术。在校外实习之前，自己曾在Coursea 上跟过吴恩达的机器学习入门课程，但是迫于实习的压力一直没有实践的机会。但是在实习过程中，自己对于机器学习的热度并没有随着实习压力的上升而减弱，反而更加强烈，这次毕业设计正是自己深入学习和实践机器学习的一个绝佳机会。

### 1.1.3 课题的意义

本课题最大的意义在于对自己的实践性意义，作为一名即将踏入工程领域的程序员，工程技术性实践将是其最大的收获。微信聊天作为一种聊天工具，与我们的生活息息相关，几乎成为了我们生活不可分割的一部分。据微信发布的用户数据显示，微信日登陆用户达9亿多，人均每天花费在微信上的时间约一小时，也就是说我们每天花费在即时聊天上的时间大约是自己工作时间的八分之一。类似Google Assistance，Apple Siri 等，微信聊天助手是在占据大量时间的即使聊天中，自动化模板聊天，以减少用户花在这些方面的精力，这样的意义在于让自己的工作和生活更加精简。

## 1.2 国内外发展状况

中国的即使聊天的工具虽然起步晚，但是后起勃发，中国的即使聊天也处于领先水平。当然国外也有很多优秀的即使聊天工具，例如Whatsapp，Facebook Messenger等。值得注意的是在Facebook 2017 F8 大会上，Facebook 发布了Messenger Chat Extensions，其中包括 Smart Replies。Smart Replies 是Facebook Messenger 内置的AI功能，它提供了学习用户常用的问题回答，来帮助用户自动回复一些常用回答。Facebook Messenger 的目的旨在将自动化的服务带给小微企业，给予其提供7\*24小时不间断服务。Smart Replies 是自动从公司的主业采集信息，识别用户的问题，然会进行回答。

国内的微信和QQ是主流的即使聊天工具，占据了大量的市场份额。QQ没有相关助手的扩展，其更核心的业务是提供更加娱乐性的功能。相对与QQ，微信提供更加简洁即使聊天体验，虽然都是腾讯的产品，其给与客户的体验截然不同。微信提供了助手相关的扩展，例如微信公众号。在微信公众号内，微信公众号的运营者可以提供给用户定制化的服务，包括定制化的聊天助手等，但是还没有针对微信用户的个人聊天助手。

# 第2章 方案论证

## 2.1 方案选择

1．方案一 前端Vue + 后端 SSM + 机器学习 Tensorflow + 微信接口 Itchat

提出课题后不久，就开始着手整个方案的选择相关事宜。基于自己的技术栈，前端选择Vue 框架。Vue 是一套用户快速构建用户界面的渐进式框架，搭配其他Vue生态圈的框架，可以构建用户友好的前端。SSM 是Spring MVC，Spring 和 Mybatis 的缩写，也是Java 后端主流解决方案之一，与 SSH ( Struts，Spring 和 Hibernate )相比，SSM具有更好的安全性和更加灵活的设计。Tensorflow 是Google Brain 开源的机器学习框架。Tensorflow 可以被用于语音识别，图像识别和深度学习等多项领域，并且可以运行在多种平台。Itchat 是Github 上面开源的一个微信聊天接口包装库，微信源生并不支持将用户的聊天开放的接口，这个开源库只是包装微信Web为API，并暴漏为Python的API。结合以上所述的Vue，SSM，Tensorflow和Itchat 理论上可以实现该课题的主要内容。

2．方案二 前端React + 后端 Spring One Stop + Tensorflow + Itchat

与上的一种方案相比，前端改用了 React ，后端改用了Spring One Stop。React 是Facebook 开源的一个前端框架，与Vue相比，React 更加适合构建单页面Web APP， 同时React 的生态圈更加丰富，搭配Redux 可以很好的管理单页面数据流。Spring One Stop 是Spring 生态圈的一个很好体现，Spring 拥有完备的生态圈，提供了Spring Boot， Spring Data JPA等。如果没有特殊需求，业界更加希望采用Spring One Stop 解决方案，这样可以让后端的每一个子模块更加无缝衔接。

3．方案三 前端React +后端 Python Flask + Tensorflow + Itchat

与前一种方案相比，后端改用了Python Flask。Flask 是基于 Python 的一个Web 开发框架，提供了更加精简的Web开发核心，通过扩展来显示的添加其他功能，而不是在内设在开发框架内部。Python 作为机器学习领域的主流语言之一，具有先天的优势。Tensorflow 虽然底层使用C++ 开发，但是应用层API暴漏的是Python接口，同时Itchat 也是暴漏的Python API。后端采用基于Python 的Flask Web 框架，可以使得后端，Tensorflow 和 Itchat 更好的无缝集成。

## 2.2 可行性分析

### 2.2.1 经济可行性

在2.1方案选择里所论述的三种解决方案，所有涉及的技术都是开源技术，都已开源协议的形式开源，在不商用的情况写不涉及授权问题。同时开发只需要个人笔记本电脑一台，可以连接网络即可。结合以上论述的内容，本课题基本不涉及大量经济投入，经济可行。

### 2.2.2 技术可行性

如2.1方案选择里所述的方案，开发本课题有多种选择方案。后端数据库所有方案统一采用Mysql 开源数据库，采用开源操作系统Ubuntu和社区版的IntelliJ IDEA。在以前的学习过程中，有基于Mysql，Ubuntu 和 IDEA 的开发经验，同时在校外学习的过程中有相关敏捷开发，CI/CD等实习经验，可以将其思想引入到本项目，以减少项目在设计实现过程中遇到的问题。在Ubuntu 操作系统的社区中有很多开源软件，可以提供基于完备的工具集合，开发中需要用到的工具基本都可以在这里找到，例如版本控制器Git，虚拟网络VPN，容器Docker等。同时作为开发者，使用Ubuntu开发，可以很好的将开源软件整合到一个平台，大多数开源软件都会提供一个Linux 发行版的二进制包，这样即增加了项目的技术可行性也有利于后期将项目使用Docker 部署到云主机。综合这些论述，不存在较大技术阻碍，技术可行。

### 2.2.3 操作可行性

本项目暴漏给微信用户的是一个单页面的Web App，用户只需扫码或则在微信里面输入相应的网址即可跳转到该Web App。用户所有相关操作都只需在该Web App 中操作。绑定微信账号到系统，用户只需扫描系统呈现的二维码，授权登陆即可。其他相关功能，用户只需同使用其他App一样的习惯即可上手本Web App。同时本项目结合了CI/CD 采用 Jenkins 和 Docker 持续集成项目，用户无需等待项目的发布。结合以上论述的几点，本项目无复杂用户不友好操作，操作可行。

### 2.2.4 法律可行性

纵观以上论述的技术，大多数技术都是出自开源社区，版权通过开源协议的形式授权，例如：React 是Facebook 以MIT 开源协议授权的一个前端开源框架，及被授权人有使用，修改和合并等权力。其他少数商业软件都是采用的社区版本，社区版本的商业软件是无需授权即可使用。同时本课题没有介入微信任何相关业务，只是引用包装了微信客户接口的第三方库来实现接口服务。且本项目为综合实践项目，不存在商业目的。综合以上几点，无严重法律问题，法律可行。

## 2.3 方案确定

方案的确立一波三折，从最开始的Vue+SSM+Tensorflow+Itchat 到 React+SOS+Tensorflow+Itcaht 最后到 React+Flask+Tensorflow+Itchat。每一种方案都有各自的好处，都有值得选择的理由。第一种方案，更像是传统互联网项目解决方案，具有普遍性。第二种方案后端技术更加契合，前端采用更加成熟的React 使得项目在理论上更加具有可行性。最后一种方案是采用Python Flask 这种方案主要是考虑到 Tensorflow 和 Itchat 都是暴漏Python API，如果采用 Flask 作为后端框架，这将使得后端跟其他服务能够更加紧密的衔接。

综合以上论述，最后采用方案三（React+Python Flask+Tensorflow + Itchat）。数据库选择Mysql，开发操作系统选择Ubuntu，开发IDE 选择社区版的 IntelliJ IDEA 和 Pycharm，版本控制器选择 Git，并且将代码托管到GitHub，部署环境选择 CentOS 搭载 Docker 和 Jenkins，敏捷看板采用Github 提供的Project。

# 第3章 需求分析

## 3.1 功能需求分析

本项目包含五大模块：微信账号绑定模块，聊天内容识别模块，聊天内容发布模块，聊天管理模块，深度学习算法设计模块。

1. 微信账号绑定模块

微信用户初次进入该系统，需要将微信与该系统绑定，同时授权该系统登陆微信以获取用户聊天记录。主要操作流程为：用户进入系统，系统展示二维码，用户扫描二维码并授权登陆，系统与微信账号绑定。

1. 聊天内容识别模块
2. 文字聊天内容识别：当微信好友发送文字内容给系统时，系统识别为文字内容，并将文字内容交由上层算法模块。
3. 语音聊天内容识别：当微信好友发送语音内容给系统时，系统接口将语音转换为文字，并将文字内容交由上层算法模块。
4. 文本文件内容识别：当微信好友发送文本文件内容给系统时，系统将文本文件内容解析为文字，并将文字交由上层算法模块。
5. 聊天内容发布模块
6. 文字内容发布：当接受到上层算法模块的文字内容时，将文字内容发送给指定的微信好友。
7. 文件内容发布：当接受到上层算法模块的文件别名内容时，将指定的文件发送给指定的微信好友。
8. 聊天内容管理模块
9. 上传常用聊天内容：通过文本文件上传常用聊天内容，通过该内容训练聊天机器人。
10. 删除常用聊天内容： 删除以前上传的聊天内容，并且从新训练聊天机器人。
11. 查询常用聊天内容： 通过别名查询常用聊天内容。
12. 列出聊天记录： 通过列表的形式列出聊天内容记录列表。
13. 算法设计模块

（1）聊天机器人算法：设计和实现聊天机器人，可以被动回复文本。

## 3.2 性能需求分析

系统的反应加载速度应尽可能的快，作为单页面Web APP 在系统第一次加载的过程中，系统会将所有的资源加载完，这样会导致第一次加载的时候稍微缓慢，所以需通过响应的技术减少第一次加载的时间，例如：资源压缩，JS代码压缩等，同时需要被动回复的时间在用户接受范围内。后端通过Restful API 给前端暴漏接口，这就需要后端服务的响应时间不能太久，如果超过限定的超时时间，前端应该给与超时的友好提示，但是尽量避免这样的情况发生。

## 3.3 系统数据流图和数据字典

1．根据需求得到如下图所示的顶级数据流图



图3-1 系统顶级数据流图

2．对顶级数据流图细化得到如下二级数据流图