



**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，响应变化等特性的服务。

**关键词：**RNN，Tensorflow，React，Flask，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，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)