

程序设计 II 实验报告

李骋昊 PB20081583

吕凯盛 PB20081590

季严彪 PB19081595

I. 需求描述

本次实验需要实现一个对话机器人，应该满足基本的需求有：对输入的对话有对应的输出，同时回复的速度不应过慢；图形界面对用户友好，便于操作；对输入语句的回复相关性尽量高。

II. 功能实现

A. 技术栈

基于高内聚、低耦合的基本原则，将程序分为前端和后端两个部分：前端接受用户输入，并将回复输出，而后端负责产生回复。经过仔细的调研，最终前端采用 nodejs+websocket+html 的方案，实现了图形界面和用户接口；而后端采用基于 intent 的神经网络模型，使用 Python 实现。nodejs 和 Python 进程之间通过 socket 通信。

B. 模块划分

基于前后端对做更详细的模块划分：前端包含图形界面模块、socket 通信模块以及 html 网页；后端包含 socket 通信模块、神经网络训练模块、模型导入模块和基于输入和模型的输出模块。

C. 实现详解

a) 后端：intent 即文本意图，通过了解文本意图机器人可以迅速对输入进行分类，并进行对应的输入。本次实验采用神经网络为

Keras 中简单的序贯模型，即最简单的线性、从头到尾的结构顺序，不分叉，通过多个网络层的线性堆叠构建出深度神经网络，并采用 relu 函数作为激活函数。训练数据集为人工标注的常见语句及其意图，输入时进行词干化预处理，最终输入为词袋及标注意图。训练完成的模型实际上可视为一个简单的词袋分类器，将输入语句词袋化之后根据相似度将其分类，并输出预置类别的回复。

b) 前端：WebSocket 是一种在单个 TCP 连接上进行全双工通信的协议。WebSocket 使得客户端和服务端之间的数据交换变得更加简单，允许服务端主动向客户端推送数据。在 WebSocket API 中，浏览器和服务端只需要完成一次握手，两者之间就直接可以创建持久性的连接，并进行双向数据传输。本次实验使用 nodejs 建立一个 websocket 服务器，并通过 JavaScript 产生动态网页，以支持动态的消息显示。网页端获取用户消息之后，通过 socket 将消息发给 8001 端口的 Python socket 服务器，Python socket 服务器收到消息之后，传给后端，并获取对应回复。回复同样通过 socket 发回给 nodejs，nodejs 通过 websocket 协议在网页上动态显示回复语句。

III. 运行结果

以下为与机器人进行一些基本对话的结果图：

由结果可知，该机器人具有基本的对话能

力和一定的话题扩展能力，满足了基本的需求；同时对话延迟控制在毫秒级别，反应灵活迅速；对于用户的意图推断较为准确，回答能够贴近主题。

IV. 总结

本次实验为程序设计 II 的综合性实验，很好的融合了字符串处理、socket 编程、软件工程等知识，比普通实验更贴近实际，让同学们更能体会实际工程、生产中面临的挑战，让知识跳出课本，成为实际体验的一部分。就本组而言，nodejs 和神经网络都是平常未接触过的全新技术，这次试验让我们有机会探究新技术同时实践所学知识。本次实验不仅锻炼了我们的编程能力，更锻炼了我们的团队协作能力、信息检索与学习能力、遇到困难解决困难的恒心与毅力。