

大作业结题报告

课 题 名 称 ： 高 级 软 件 工 程

报 告 题 目 ： 多模型智能聊天机器人设计与实现

学 生 姓 名 ： 邓浩宇 徐早辉 吕玮智

班 级 ： 高级软件工程2班

指 导 教 师 ： 罗铁坚

完 成 日 期 ： 2021 年 12 月 26 日

目录

[一、项目背景](#_Toc995654568)

[1.1 背景介绍](#_Toc773746952)

[1.2研究现状](#_Toc1351539679)

[1.3技术路线](#_Toc1392850634)

[二、 需求分析](#_Toc2068853338)

[2.1 用户故事](#_Toc1310323189)

[2.1.1 普通用户](#_Toc157037538)

[2.1.2 管理员](#_Toc72499003)

[2.1.3 用例图](#_Toc867515572)

[2.2 功能需求分析](#_Toc1067739121)

[2.2.1 用户管理模块](#_Toc1118052315)

[2.2.2 基础对话模块](#_Toc623346955)

[2.2.3 其他模块](#_Toc1167042619)

[2.2.4 功能模块图](#_Toc1517149482)

[2.3 非功能需求分析](#_Toc1658003143)

[2.3.1 安全性](#_Toc311020929)

[2.3.2 并发性](#_Toc353556905)

[2.3.3 兼容性](#_Toc143651086)

[三、 系统设计与实现](#_Toc572183174)

[3.1 前端设计](#_Toc250834152)

[3.1.1前端结构](#_Toc259193603)

[3.2后端设计](#_Toc1170049505)

[3.2.1 数据库结构](#_Toc514274956)

[3.2.2 后端服务器设计](#_Toc1944989964)

[3.2.3 对话生成系统模块架构](#_Toc450250314)

[3.2.4 对话生成流程](#_Toc1772139017)

[四、 对比测试与自动化测试](#_Toc889758476)

[4.1 对比测试用数据集](#_Toc1242072071)

[4.2 对比测试——自由对话](#_Toc1964248457)

[4.3 对比测试——功能问答](#_Toc2005195115)

[4.4 pytest单元测试](#_Toc853425434)

[4.5 仓库管理和集成测试](#_Toc477990925)

[五、 遇到的突出问题与解决方案和思路](#_Toc2004636695)

[5.1 突出问题](#_Toc57995082)

[5.2 对应解决方案或思路](#_Toc1913251083)

[六、 项目部署与组员分工](#_Toc1738305450)

[6.1 项目进度](#_Toc1332164362)

[6.2 组员分工](#_Toc21928512)

[6.3 项目发布](#_Toc1332797547)

# 一、项目背景

## 1.1 背景介绍

随着人工智能、互联网等技术的发展，聊天机器人应用多重领域，在电信、旅游、医疗航空、金融等领域均有涉及，效果明显。如在网购、电子商务、电信等服务要求高，访客数量多的领域，需要大量客户时刻等待用户的询问，而部署对话机器人即可在满足大部分用户基础的服务要求上，极大的减少了客服的压力，而在医疗保健、体育、教龄前育儿等领域，可以提供基础的帮助服务，提升用户的幸福感。

人工智能技术突破了聊天机器人原有的技术瓶颈，并且在实践中证明，聊天机器人不仅能够为企业节省一大笔人力成本，并且能够极大的提高工作效率，同时也能服务于个人用户如智能家居、智能宠物机器人等方面，因此国内外多家企业纷纷布局对话机器人行业，未来此行业必会一片欣欣向荣，因此本组选择对话机器人作为我们的软件工程大作业项目。

## 1.2研究现状

在全球角度来看，因为国外在 AI 领域和聊天机器人及问答系统方向的研究起步较早，也发展出了不少比较成熟的问答系统，例如 Google 公司的Google Home，Microsoft 公司的小冰，Apple 公司的 Siri，Facebook 公司Messenger 等。这些大体量的 IT 企业研发的 AI 器人，都借助着其公司在大数据（BD）、自然语言处理 (NLP)、机器学习 (ML) 和深度神经网络 (DNN)等方面的优势技术，利用积累的庞大和较高质量的语料库，训练出了具有相当实用价值的智能问答机器人模型。以此实现了超越一般简单人机对交互，而基于自然语言环境的智能问答，为用户提供了新的便利与乐趣。

对比外国，我国在 AI 聊天机器人方面的投入和研究水平上都有着不小的落后，总体发展成果相较美国来说还有相当的差距。但是还是有不少高校和研究机构在此领域有显著成绩，研究水平较为突出的有清华大学、中国科学院计算所、香港大学和哈工大等。而国内在此领域的研究主要集中在对自然语言处理工具的开发，比如哈尔滨工业大学的 HIT 工具以及台湾国防大学的 CQAS 中文问答系统等。而由于中文的语句特点，不像英语等语言一样词之间有自然间隔，对于中文的分词具有更高的难度。

## 1.3技术路线

LSTM(Long Short Term) 网络是一种循环神经网络（RNN）特殊的类 型，可以学习长期依赖信息。LSTM已经在科技领域有了多种应用。基于LSTM 的系统可以完成翻译语言、控制机器人、图像分析、文档摘要等任务。 本文的生成式智能问答系统就采用了 LSTM 网络组成 seq2seq 模型，进行自然语言的理解和生成。LSTM 单元结构如图所示：

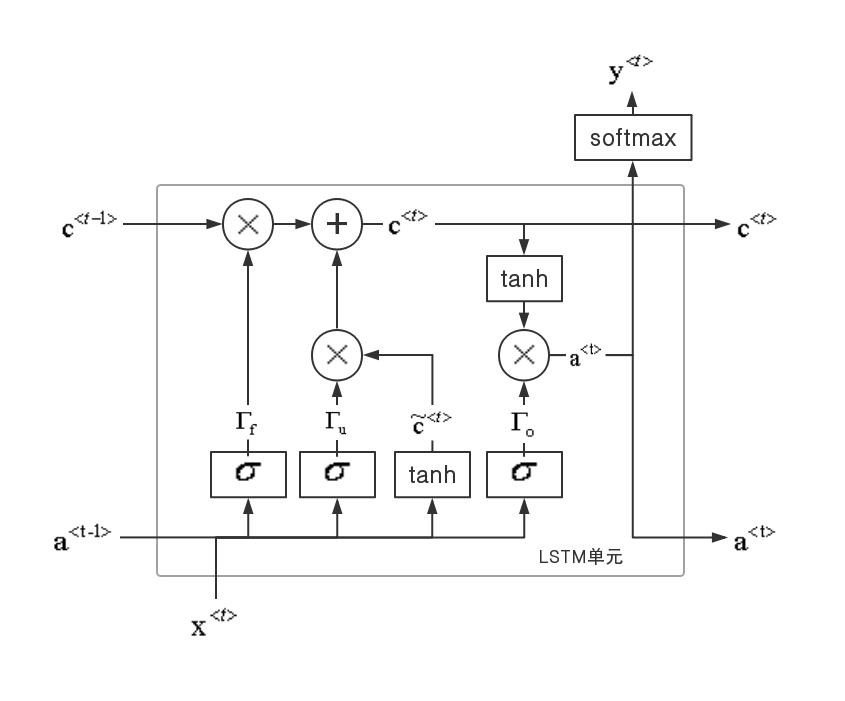


图 1: LSTM 单元结构图

而GRU单元是一种类似于LSTM的简化单元，更加易于训练。

Seq2Seq 模型是一种输入和输出都是序列的模型，由编码器和解码器组成，而编码器和解码器又是由 LSTM 或者 GRU 单元相互连接组成的，如图所示，图中采用的是 LSTM 单元。与编码器不同的是解码器的第一个计算单元获取了编码器产生的中间语义向量，同时每个时间步都有对应的预测输出向量，这些向量就代表了依次输出的词语。同时，从第二个时间步开始，LSTM 单元的输入采用了上一个计算单元的输出向量，这也是为了保证输出序列内部的前后关系正确。最后当整个解码器预测输出一个停止标签时，整个解码过程结束。

下一层的输出值作为上一层的输入值进入到上一层的 LSTM 神经元，这就形成了一个多层的 LSTM 网络结构。图中从左下到左上斜向上排列的三个神经元分属与三层网络结构，在计算上也是由下至上地进行计算，而第二时间步的第二层神经元则需要等待上一个时间步和上一层计算完毕的数据输入才能进行计算，以此类推之后的计算方式。而当多层 LSTM 网络作为解码器时最上层上一时刻的神经元的输出同样像单层 LSTM 网络解码器一样，需要输入到下一时间步的最底层神经元，从而完成整个序列的预测。

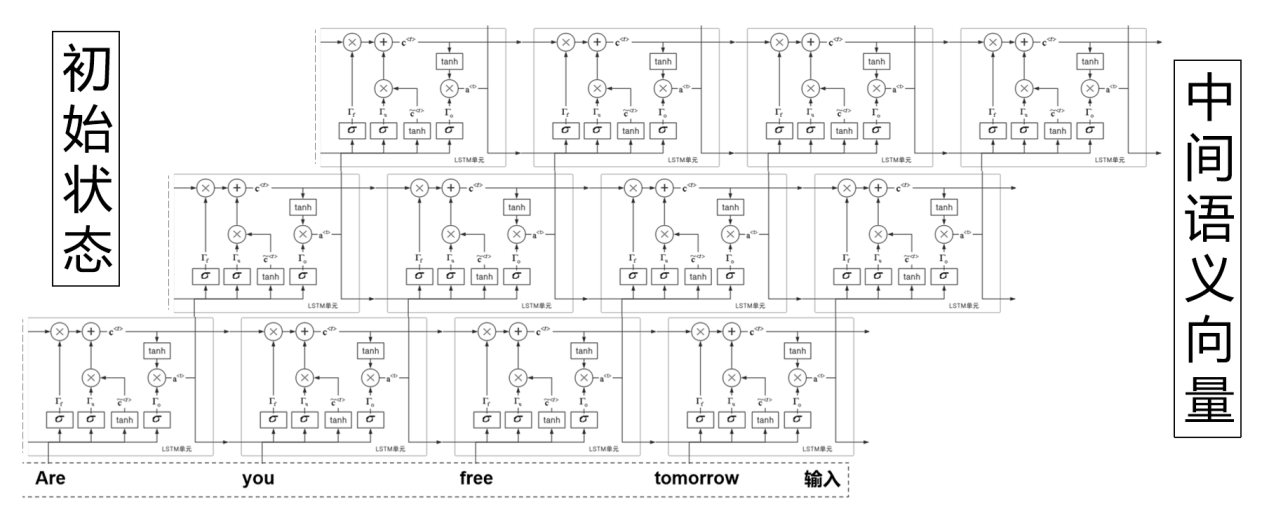


图 2: Seq2Seq 模型编码器结构图

# 需求分析

## 2.1 用户故事

### 2.1.1 普通用户

游客通过注册成为普通用户。普通用户又分为三类：一类是对对话机器人需求较低，只使用非常基本的查询功能（例如算数、查天气等）的用户； 第二类是愿意与聊天机器人对话，产出丰富语料的用户；第三类是更强需求的用户，除了以上需求外，还有与获取资讯、进行特定主题聊天的需求。

### 2.1.2 管理员

拥有管理员权限的用户成为管理员。管理员负责管理用户信息、查看修改用户权限、维护 NLP 模型、维护系统功能。

### 2.1.3 用例图

下图是系统用例图，涉及角色用户和管理员，涉及活动：登录注册、 管理模型、管理用户等。

图示

描述已自动生成

图 3: 系统用例图

## 2.2 功能需求分析

### 2.2.1 用户管理模块

用户管理模块包括用户的登录注册功能，用户信息修改功能。用户管理 模块旨在增加用户粘性，为用户提供更好服务，记录并管理用户信息，除去基本信息外，比如对用户权限的管理，对人机对话信息的管理。

### 2.2.2 基础对话模块

对话模块基于不同语料数据集训练出来的机器人，可以针对用户输入 的语句进行日常对话。可以通过切换不同模式，切换不同的语料数据集下不 同聊天特点的的机器人与用户进行对话。

### 2.2.3 其他模块

其他模块包括查新闻和查天气、查询运势功能。本模块旨在为用户提供方便，当有相关需求时能类似于智能机器管家，通过对话方式快速得到用户所 需要的信息，或查询所需信息而不用跳转到其他页面，节约了用户的时间。

### 2.2.4 功能模块图

对话机器人包括三个主要模块：用户管理模块、基础对话模块、其他功能模块，每个功能模块下还有细分。

图示

描述已自动生成

图4: 功能模块图

## 2.3 非功能需求分析

### 2.3.1 安全性

平台记录了大量用户信息，包括用户个人信息和用户与机器的对话信 息，其中包含众多用户的个人隐私。因此平台亟需具备安全性保障用户不被 窥探，进而增加用户的信任度和用户粘性。

### 2.3.2 并发性

平台可能遇见大量用户同时使用平台聊天机器人功能的应用场景，此时 需要高并发维持系统正常运行，使得每个用户可以较流畅地使用平台功能。

### 2.3.3 兼容性

对话机器人平台需要集成 Linux 系统下的训练板块和在主流平台 Web 端的显示板块。因此平台要兼容自适应各种平台系统，在各个平台系统下都 能完成预期任务。

# 系统设计与实现

## 3.1 前端设计

### 3.1.1前端结构

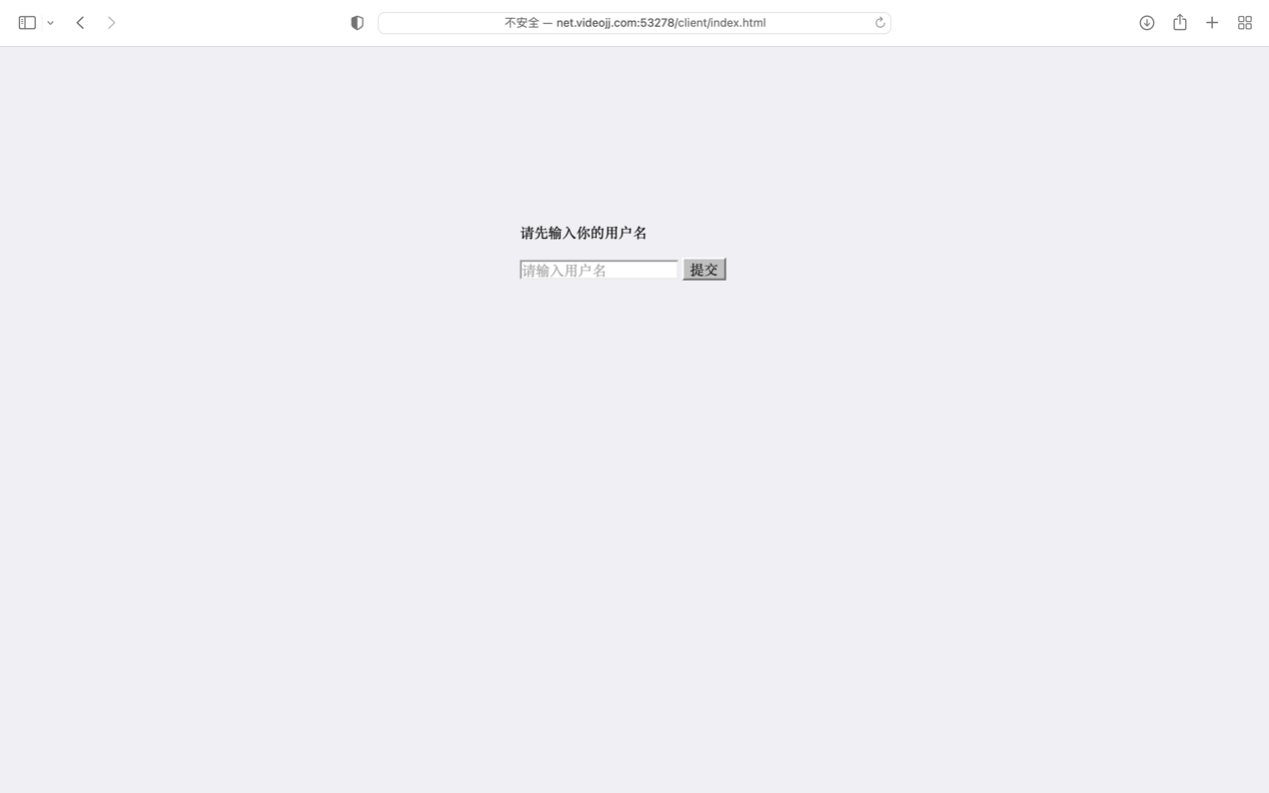


图5: 前端界面图

前端部分包含html，css的界面设计和javascript的客户端脚本设计，首先是一个登录界面输入用户名后用户名会经由javascript发送给服务器完成数据库的录入。

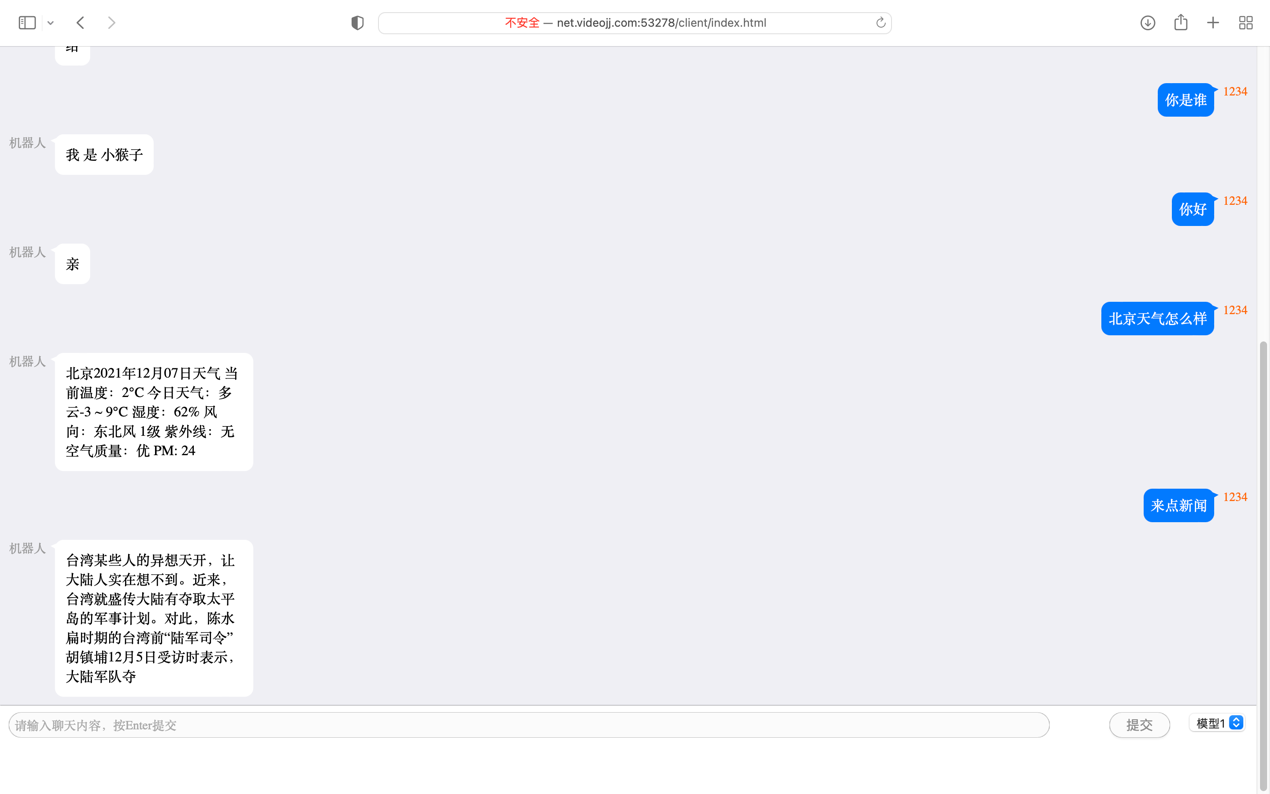


图6: 对话效果图

聊天的主界面，包含聊天记录框，聊天输入框，一个退出按钮和一个发送按钮，在聊天记录框可以看到和机器人的聊天内容，在输入框输入内容后点击发送将会发送给服务器，经处理后的回复返回客户端并显示在聊天记录框。点击退出按钮即可退出登录。

## 3.2后端设计

### 3.2.1 数据库结构

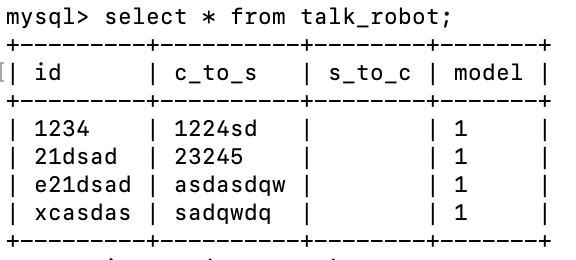


图7: 数据库结构图

使用Mysql作为服务器端的数据库，存储用户id、对话信息以及模型选择信息。由用户发送给服务器的语句以及对话系统处理生成的结果均存放在数据库中。数据库是系统的数据中转站，连接着用户端与对话系统。

### 3.2.2 后端服务器设计

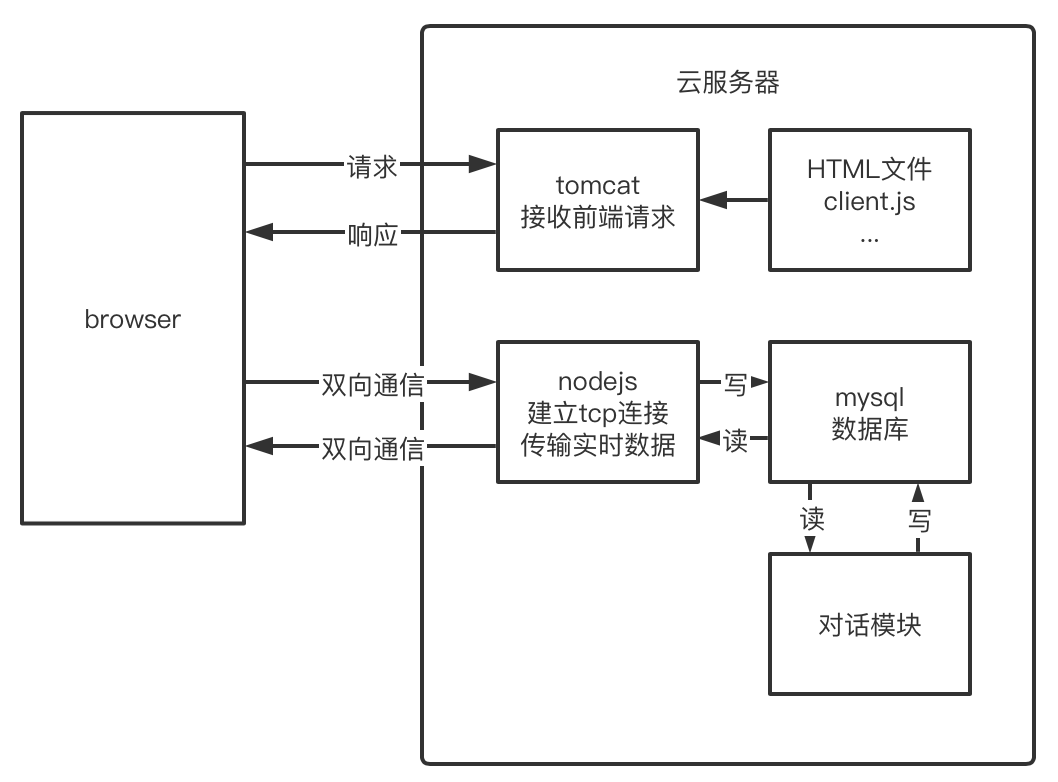


图8: 后端模块图

后端的服务器选择tomcat、nodejs这两种轻量级服务器。Tomcat作为web服务器用于传输服务器上的文件到客户端的浏览器上，Nodejs作为应用服务器使用websocket建立与客户端的tcp连接实现服务器与客户端之间的双向通信，并且实现对数据库的读写操作。

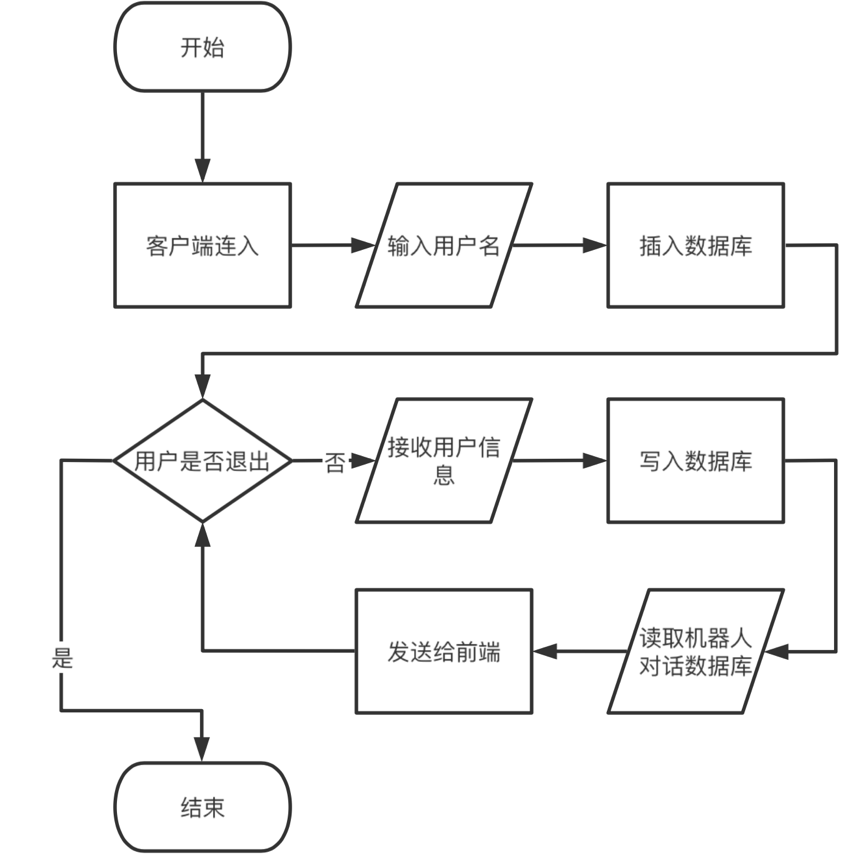


图9: 后端流程图

### 3.2.3 对话生成系统模块架构

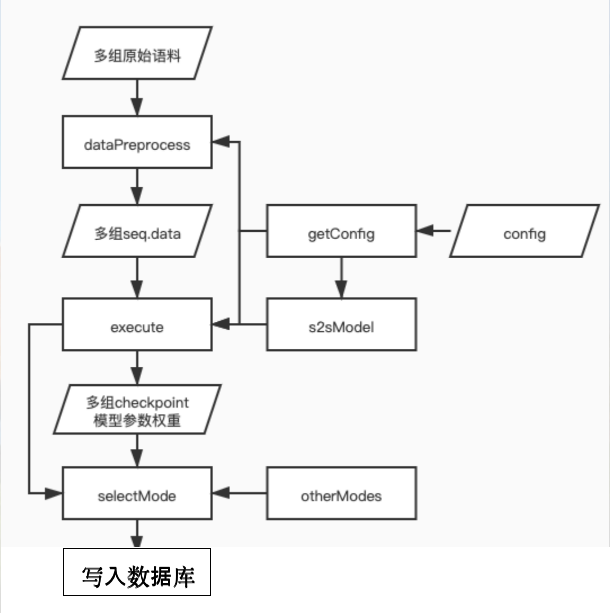


图 10: 对话生成系统模块架构图

多组原始语料数据：包括多种语料，用于训练不同风格和功能的对话机器人。

dataPreprocess 模块：用于数据预处理。该模块读入原始语料，对语料进行分类和分词处理。

多组 seq.data 数据：是多组原始语料经过预处理后的结果，开始训练时被execute 模块读取。

execute 模块：用于训练 seq2seq 模型和针对输入语句进行预测。该模块首先进行数据集创建，将seq.data 处理成可以直接输入神经网络模型训练的数据集，具体操作包括读取数、给语句增加标志、将语句向量化。同时定义训练循环，在循环中输出必要提示信息，保存模型权重。最后定义预测函数，负责计算预测结果。

config 文件：用于保存模型超参数。

getConfig 模块：用于dataPreprocess、execute、s2sModel模块，读取 config 文件中的超参数。

s2sModel 模块：用于定义 seq2seq 模型结构。

多组 checkpoint 模型参数权重数据：是由多组原始语料训练出来的对应模型参数权重。

otherModes模块：定义了生成回复用户语句的各种模型和功能函数。提供多种更有功能性的回复方法。

selectMode 模块：用于自动判断用户输入的语句应该使用何种模型或功能函数进行回应。

### 3.2.4 对话生成流程

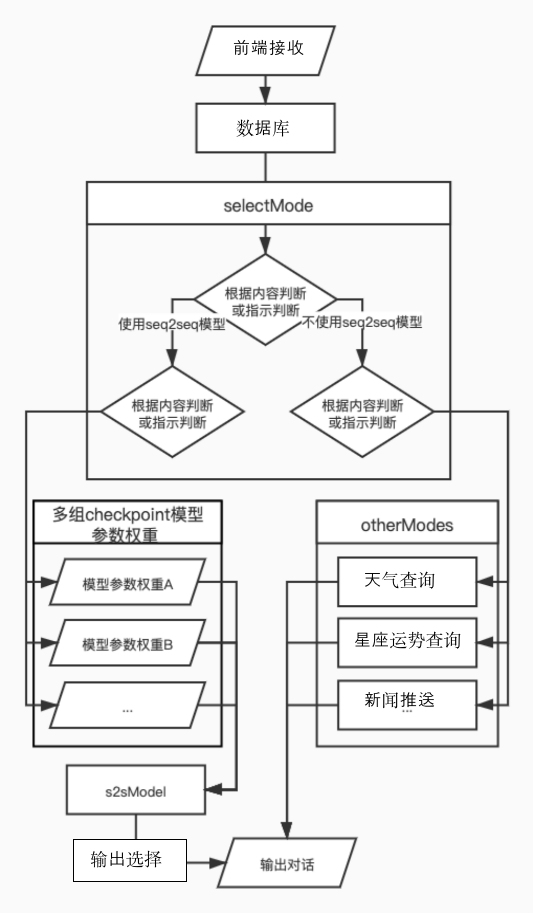


图 11: 对话生成流程图

用户在web端输入语句后，随即写入云服务器中的数据库，随后传入selectMode模块，该模块根据语句内容自动选择使用的模型。若使用seq2seq模型，则在多组checkpoint模型参数权重中选择对应模型的最新权重，随后excute模块的预测函数使用加载对应权重的seq2seq模型计算预测结果，得到输出对话。若不使用seq2seq模型，则在otherModes中选择对应的回复模式，该模块处理后得到所有seq2seq模型的输出语句，再经过情感分析评价，输出其中最合适的语句。

# 对比测试与自动化测试

## 4.1 对比测试用数据集

测试数据集共47句问句，其中35句自由聊天与训练语料相关，12句是查询信息的功能性问句。

图12、13是自由聊天问句，黄色标注的问句是多模型（本项目）回答效果明显优于单一模型（原开源项目）效果的问句，红色标注相反。图14问句为查询信息的功能性问句。

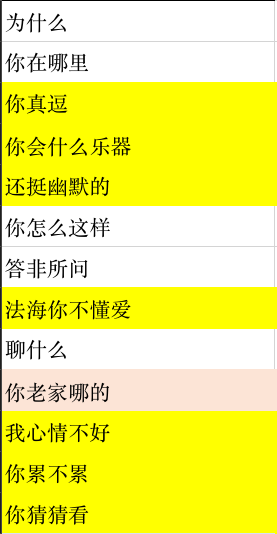


图 12: 自由聊天测试问句1

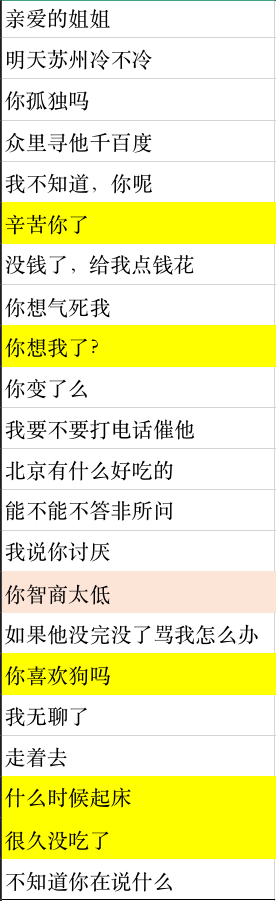


图 13: 自由聊天测试问句2

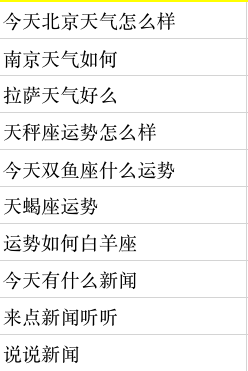


图 14: 查询信息功能测试问句

## 4.2 对比测试——自由对话

图15为多模型输出效果明显好于单模型结果（原开源项目）的展示，占测试集中的12句，占自由对话部分的34%。

下图16为单模型结果（原开源项目）明显好于多模型输出效果的展示，占测试集中的2句，占自由对话部分的6%。



图 15: 多模型输出效果明显好于单模型的结果展示

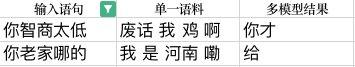


图 16: 单模型结果（原开源项目）明显好于多模型输出效果的展示

## 4.3 对比测试——功能问答

图17为查询信息的功能性问句测试结果，多模型（本项目）回答效果明显优于原开源项目，增加了天气、运势、新闻的对话询问功能。

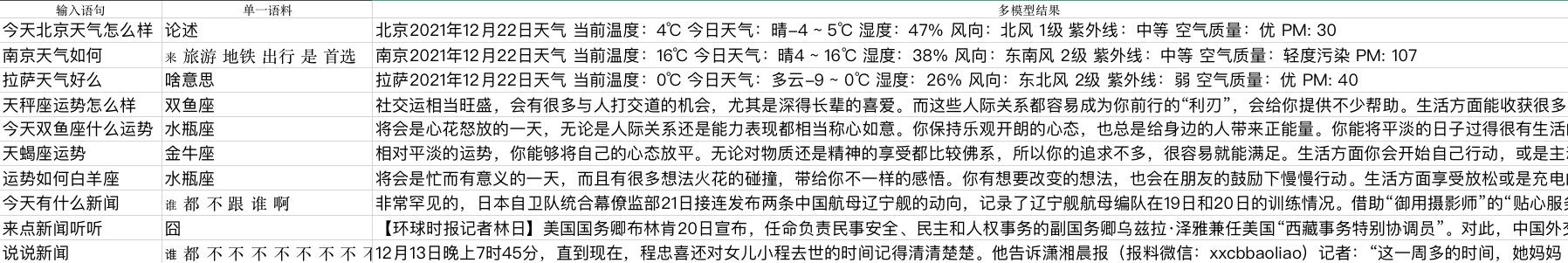


图 17: 查询信息的功能性问句测试结果

## 4.4 pytest单元测试

本项目使用的测试工具是pytest自动化测试框架，测试用例编写在项目根目录下的test\_start.py中，配置文件为pytest.ini，使用pytest -s -v进行测试。测试包括的模块有数据预处理模块、seq2seq模型训练模块、模型选择模块。测试内容主要包括函数测试和关键变量检查。本项目通过了所有测试用例的测试，结果如图18所示。

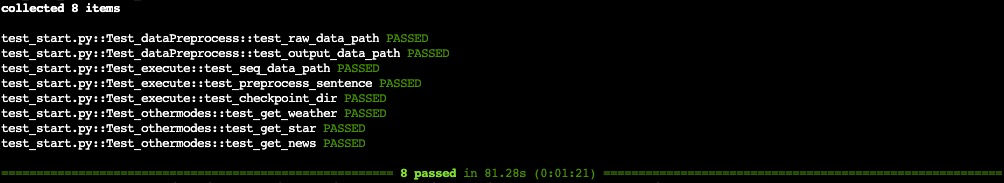


图 18: pytest自动化测试结果图

## 4.5 仓库管理和集成测试

在项目的开发过程中我们主要使用线上的JupyterLab进行协同开发。在主体部分完成后，将代码放在了托管在了Github进行了代码的仓库管理，并继续进行持续的开发。除此以外，还使用了Github Actions Runner进行了集成测试与CICD。

电脑屏幕截图

描述已自动生成

图19：代码仓库管理

图形用户界面, 文本, 应用程序

描述已自动生成

图20: Github Actions Runner进行集成测试与CICD

# 遇到的突出问题与解决方案和思路

## 5.1 突出问题

1. 跑训练用的服务器虽然显卡更好，但是无法开放与前端传输数据的接口，新租用的云服务器提供对外的端口号有限，需要进行相关配置√
2. nodejs作为后端web服务器是单线程，且是完全的异步读写，对事件处理流程的编写造成的了因扰，不能做cpu密集型操作，较Java、PHP对多用户支持较差。×
3. 对特定主题的识别不够精确精准，有时会出现错误×
4. 新闻调用的公开api不稳定，可能失效√

## 5.2 对应解决方案或思路

1、云服务器仅给了两个对外访问的端口号，一个是2167，另一个是53278，分别用于tensorbaord和jupyterlab的访问，因此需要将这两个进程先杀死并且不再自启动。开启后端服务器时tomcat要绑定在6006端口号，他是固定映射在对外端口53278上，nodejs要绑定在8888上，他是固定映射在对外端口号2167上，具体原因未知，可能是NAT端口映射或是内网穿透技术导致的。

2、nodejs全是异步调用和非阻塞I/O，Node 会将I/0请求封装成一个对象；将这个请求对象推入到 I/O 线程池等待执行。使用Promise 和async回调函数实现同步读写。但对多用户的支持还需进一步研究相关知识。

3、使用正则表达式进行匹配特定主题关键字，好处是方便快捷。但是对于一些特殊的问法则可能出现正则匹配不上的问题，这个问题并不好解决，需要一些复杂的人工智能算法理解文本本身。

4、在调用API时有时提示SysCallError: (10054, 'WSAECONNRESET')，经过排错发现这并不是网络连接问题或系统问题，经过多个API的测试，最终换了另一个内容主题但更加稳定的接口。

# 项目部署与组员分工

## 6.1 项目进度

2021.09.24–2021.10.09 修改开题报告和整体设计

2021.10.10–2021.11.10 开源项目调研、测试、针对项目目标进行改造

2021.11.10–2021.11.17 多模型切换功能实现，前后端实现、自动化测试

2021.11.18–2021.12.01 云服务器迁移，整体运行测试，debug

2021.12.01–2021.12.8 文档、PPT撰写和准备演示视频

## 6.2 组员分工

邓浩宇：前期调研，开源项目调研、测试，问答部分流程整体设计，模型I/O部分的改进，运行环境搭建，单元自动化测试实现，问答效果测试数据集的建立，所有提交材料的部分编写和录制。

徐早辉：前端页面以及客户端，搭建并配置tomcat服务器和nodejs服务器，mysql数据库配置并完成相关的读写操作，所有提交材料的部分编写和录制。

吕玮智：爬虫代码(天气、星座、新闻)，模式选择代码编写，训练模型二，代码仓库管理，CICD及集成测试，所有提交材料的部分编写和录制

## 6.3 项目发布

云服务器：极链AI云

操作系统：Ubuntu

web框架：node.js tomcat

软件环境：TensorFlow2.3及相关库

硬件环境：Intel(R) Xeon(R) Gold 6129 CPU，30G ssd，TITAN Xp 1，显存11GB，内存16 GB

网站地址：http://net.videojj.com:53278/client/index.html

**6.4 代码托管**

将代码托管在平台github上，dev分支用于CI/CD，最终合并到main分支中

托管网址：<https://github.com/HungryKonata/seq2seqChatbot/tree/main>