

电子科技大学

UNIVERSITY OF ELECTRONIC SCIENCE AND TECHNOLOGY OF CHINA

学士学位论文

BACHELOR THESIS



论文题目

基于 BERT 和 BM25 的智能问答

系统设计与实现

学 院 信息与软件工程学院

专 业 软件工程

学 号 2018091620013

作者姓名 廖梓尧

指导教师 吴 劲

摘要

随着社会的高速发展，互联网规模发展的日趋庞大，人们对信息获取需求暴增。特别的，在后疫情时代，人们在医疗健康方面的需求日益增加，通常搜索引擎是人们用来满足自身对医疗信息所需的主要来源之一。而在用户提出了医疗类问题之后，国内几种常用的搜索引擎通常只会将与问答文本中有一定相关性的网页链接给予用户展示，导致无法直接有效地给出精确的回答，搜索引擎给出的结果对于用户来讲就失去了其权威性、准确性和有效性。

在大数据与人工智能的时代，以及国家智慧医疗数字化项目推进的背景下，本文提出了一种基于 BERT 和 BM25 的智能问答系统设计与实现方案，目的是在医疗垂类下，为人们提供精确可靠的智能问答服务，文中重点研究了智能问答系统的核心算法与模块设计，同时在系统实现过程中融入了软件工程相关核心理论，最后以软件的形式来呈现该智能问答系统。本文的研究与实现过程的内容分为三部分：

首先，本文从系统架构方面进行了技术选型。提出了一个基于 BERT 和 BM25 的智能问答系统设计方案，可以精准分析用户在医疗健康领域的提问，满足了用户医疗数字化信息的需求。

另外，本文提出了一个基于 BERT 和 BM25 的智能问答算法。该算法使用的是 BERT 预训练模型以及 BM25 词袋模型，分别以语义解析的方式与字面解析的方式构建问答系统的排序层和召回层。本文提出的核心算法方案有效地提升了问句分析的能力。具体的，算法首先结合用户问句的字面分析方法，进行初步相似文本筛选，再通过语义分析方法进行最优排序，得到与用户问句最相似似的候选问句，最终在关系型数据库（MySQL）通过该候选问句查找到对应的答案。

最后，本文将结合 HTML、JavaScript 等前端技术及 Flask 等后端技术，同时融入本文提出基于 BERT 和 BM25 的智能问答算法，通过 Web 端进行实现。另外，为了保证本系统的权威性与准确性，该系统的专业知识数据来源选自“Chinese medical dialogue data 中文医疗对话数据集”。后进行系统测试，该系统交互体验良好，功能有效，能够达到人们对日常医疗保健相关问题的查询需求满足的标准。

关键词：智能问答系统，BERT，BM25

Abstract

With the rapid development of society and the growing scale of the Internet, people's demand for information acquisition has skyrocketed. In particular, in the post-epidemic era, people's needs for medical and health care are increasing day by day. Usually, search engines are one of the main sources that people use to meet their medical information needs. However, after a user raises a medical question, several commonly used search engines in China usually only show the user a link to a web page that is related to the question and answer text, which makes it impossible to give an accurate answer directly and effectively. The result will lose its authority, accuracy and validity for the user.

In the era of big data and artificial intelligence, and in the context of the promotion of the national smart medical digital project, this thesis proposes a design and implementation scheme of an intelligent question answering system based on BERT and BM25, which aims to provide people with accurate information in the medical vertical category. Reliable intelligent question answering service, the thesis focuses on the core algorithm and module design of the intelligent question answering system, and integrates the core theories of software engineering in the system implementation process, and finally presents the intelligent question answering system in the form of software. The research and implementation process of this thesis is divided into three parts:

First, In this thesis, the technology selection is carried out from the aspect of system architecture. A design scheme of intelligent question answering system based on BERT and BM25 is proposed, which can accurately analyze users' questions in the medical and health field and meet the needs of users for medical digital information.

In addition, this thesis proposes an intelligent question answering algorithm based on BERT and BM25. The algorithm uses the BERT pre-training model and the BM25 bag-of-words model to construct the ranking layer and recall layer of the question answering system by means of semantic parsing and literal parsing. The core algorithm proposed in this thesis effectively improves the ability of question analysis. Specifically, the algorithm first combines the literal analysis method of user questions to perform preliminary similar text screening, and then performs optimal sorting

through semantic analysis methods to obtain candidate questions that are most similar to user questions. Finally, the relational database (MySQL) to find the corresponding answer through the candidate question.

Finally, this thesis will combine front-end technologies such as HTML and JavaScript and back-end technologies such as Flask, and at the same time integrate the intelligent question answering algorithm based on BERT and BM25 proposed in this thesis, which will be implemented on the Web side. In addition, in order to ensure the authority and accuracy of the system, the professional knowledge data source of the system is selected from "Chinese medical dialogue data". After the system test, the system has a good interactive experience and effective functions, and can meet the standards of people's query requirements for daily medical and health care related issues.

Keywords: intelligent question answering system, BERT, BM25

目录

摘要.....	I
ABSTRACT.....	II
目录.....	IV
第一章 绪论.....	1
1.1 研究背景和意义.....	1
1.2 国内外研究现状.....	1
1.3 论文主体内容.....	2
1.4 论文组织结构.....	2
第二章 相关理论与技术基础.....	4
2.1 前端开发技术基础.....	4
2.2 后端开发技术基础.....	5
2.3 词嵌入表示技术.....	6
2.4 智能问答系统相关方法.....	9
2.5 本章小结.....	10
第三章 基于 BERT 和 BM25 的智能问答系统分析与概要设计.....	11
3.1 系统存在问题与挑战.....	11
3.1.1 系统概述.....	11
3.1.2 存在问题与挑战.....	11
3.2 可行性分析与需求分析.....	13
3.2.1 可行性分析.....	13
3.2.2 需求分析.....	14
3.3 系统概要设计.....	16
3.4 本章小结.....	18
第四章 基于 BERT 和 BM25 的智能问答系统详细设计与实现.....	19
4.1 系统详细设计.....	19
4.1.1 系统模块详细设计.....	19
4.1.2 系统算法详细设计.....	23
4.2 运行环境说明.....	30
4.3 系统功能实现.....	30
4.3.1 系统智能问答模块实现.....	30

4.3.2 系统数据处理模块实现.....	32
4.3.3 系统用户交互模块实现.....	34
4.4 本章小结.....	35
第五章 基于 BERT 和 BM25 的智能问答系统测试.....	36
5.1 系统测试.....	36
5.1.1 功能性测试.....	36
5.1.2 非功能性测试.....	38
5.2 系统运行效果展示.....	39
5.3 本章小结.....	41
第六章 总结与展望.....	42
6.1 课题完成情况总结.....	42
6.2 课题未来工作展望.....	42
6.3 软件工程职业素养认识.....	43
致谢.....	45
参考文献.....	46
外文资料原文.....	47
外文资料译文.....	48

第一章 绪论

1.1 研究背景和意义

在互联网和大数据时代下，人们对于信息需求的爆炸式增长。为了满足人们的这些爆炸式的信息需求，信息检索系统应运而生，同时随着人们的需求不断增长，信息检索系统也不断发展。其中，“问答系统”是信息检索系统的一种高级形式，它与普通的信息检索系统有一些不同：其一是完口语化的语句作为其查询方式，其二是明确的答案文本作为其返回结果。使用者面对这种系统，便会省去大把筛选搜索引擎回传链接及摘要的时间成本，这无疑对于信息检索的效率提升与成本削减有的很大帮助。同时，问答系统无论在各个领域，都能准确满足使用者的检索需求，从而省去了人工筛选信息的时间，间接地提高了社会生产力。

特别的，在后疫情时代，人们在医疗健康方面的需求日益增加，通常搜索引擎是人们用来满足自身对医疗信息所需的主要来源之一。而在用户提出了医疗类问题之后，国内几种常用的搜索引擎通常只会将与问答文本中有一定相关性的网页链接给予用户展示，导致无法直接有效地给出精确的回答，搜索引擎给出的结果对于用户来讲就失去了其权威性、准确性和有效性。

基于以上原因，本文在智能问答系统方向进行研究，利用“Chinese medical dialogue data”公开数据集^[1]构建问答知识库，提出一种基于 BERT^[2]和 BM25^[3]的智能问答系统，最终构建了一个能够满足人们对日常医疗保健领域需求的问答系统，该系统对智慧医疗数字化进程有着一定推进的作用与意义。

1.2 国内外研究现状

随着人工智能、机器学习和深度学习技术的不断发展，智能问答系统已经成为了国内外自然语言处理的热门方向。目前，主流的研究方向有三种：其一，为基于知识图谱的问答系统。它以知识图谱构建事实性的问答系统，在业界是一种比较靠谱的做法，从知识图谱中寻找答案，例如郑楚杰等人提出的 DiffKS 模型^[4]。其二，为基于阅读理解 QA，它的答案是通过对段落文本进行阅读理解得到的，又可以分成抽取式、匹配式和生成式的问答方式，例如 Minjoon Seo 等人设计的 BiDAF 模型^[5]；其三，为基于多轮交互的对话系统，它主要通过四大部分实现复杂的对话系统：SLU（Spoken Language Understanding，

对话语言理解)、DST (Dialogue State Tracking, 对话状态追踪)、DPO (Dialogue Policy Optimization, 对话策略优化) 和 NLG (Natural Language Generation, 自然语言生成)。

1.3 论文主体内容

基于上述背景, 本文的主要研究内容是基于 BERT 和 BM25 的智能问答系统设计与实现, 系统预期能够针对领域垂类下的问题进行智能问答。当前系统主要在医疗健康垂类领域进行实现, 预期目标为能够满足人们对日常医疗保健相关问题的个人需求, 有效降低医疗健康领域的人力成本, 发挥推进智慧医疗数字化信息的进程的作用。

本文具体研究内容如下:

其一, 研究基于 BERT 和 BM25 的智能问答系统的分析与概要设计:

本文设计了一款基于 BERT 和 BM25 的智能问答系统, 首先拆解出在课题任务执行的过程中存在的问题与挑战, 并针对其中具体的问题点进行详细的可行性分析与需求分析, 再基于需求分析对系统进行概要设计。

其二, 研究基于 BERT 和 BM25 的智能问答系统详细设计与实现:

本文基于系统的概要设计, 细化出了系统完整的详细设计方案。同时为了将研究内容具体实现到现实生活中, 本文结合提出的基于 BERT 和 BM25 的智能问答系统的模块详细设计与算法详细设计, 对课题项目进行落实, 最终在 web 端搭建了一个完整的问答平台。

其三, 研究基于 BERT 和 BM25 的智能问答系统测试:

本文对提出的基于 BERT 和 BM25 的智能问答系统进行相应的测试, 保证实现项目满足课题目标预期以及达到交付状态, 能够提供一个知识完备且稳定的医疗健康智能问答服务。

1.4 论文组织结构

本文一共由六章组成, 每一章的组织结构如下:

第一章, 绪论: 本章首先介绍课题的研究背景和意义, 详细阐述国内外的研究现状, 并根据当前研究介绍本文主要的方向与内容。

第二章, 相关理论与技术基础: 本章主要介绍本文研究智能问答系统涉及的相关理论与技术基础, 分别从前端开发技术基础、后端开发技术基础、词嵌入表示技术和智能问答系统相关方法进行介绍。本章内容将会为后续课题设计实现提

供理论基础支撑。

第三章，基于 BERT 和 BM25 的智能问答系统的分析与概要设计：本章首先提出了当前课题存在的问题与挑战，然后分别从可行性分析与需求分析、系统概要设计介绍基于 BERT 和 BM25 的智能问答系统。

第四章，基于 BERT 和 BM25 的智能问答系统详细设计与实现：本章主要介绍系统的详细设计与实现，首先分别从系统模块设计与系统算法设计详细阐述了系统的详细设计，接着介绍了系统所在运行环境，最后对系统的各个核心模块进行功能实现。

第五章，基于 BERT 和 BM25 的智能问答系统测试：详述系统的关键功能测试过程与系统非功能测试过程，最后对系统的具体运行效果进行了展示。

第六章，总结与展望：本章对论文的主要完成工作进行了总结，并对存在的问题进行分析，接着提出了对课题未来工作的展望，最后归纳了个人在课题研究过程中对软件工程职业素养的认识。

第二章 相关理论与技术基础

本章将着重对基于 BERT 和 BM25 的智能问答系统设计与实现中涉及到的相关理论与技术基础进行较为详细的介绍。具体的，2.1 节将会介绍前端开发技术基础，2.2 节将会介绍后端开发技术基础，2.3 节将会介绍词嵌入表示技术的相关理论，2.4 节将会介绍智能问答系统在实现层面的相关方法，2.5 节将会对本章进行一个总结。

2.1 前端开发技术基础

前端开发（Front-end development）技术涉及到 HTML、CSS、JS、JQuery。本节将会对提到的这些技术进行一一介绍。

HTML（HyperText Markup Language），中文全称是“超文本标记语言”。它是一种标记语言，开发者一般用 HTML 创建 Web 网页的“骨架”。HTML 可以结合文本和它自身的相关信息，从而展现出开发者想要的的数据与网页结构。首先，其 HTML 元素在网站构建中的地位如网站的基石一般，通常在网络浏览器运行的过程中，开发好的 HTML 文件会被浏览器在某个阶段读入，同时浏览器会通过渲染的方式来使得网页能够展示出相应的数据；其次，HTML 语言拥有一个网站的结构语义的功能，也正是这个原因，使之的定位成为了一种标记语言；另外，HTML 也额外提供了可以用于创建表单的功能，同时也可以允许嵌入图像等基础性的操作；再者，它被用来结构化如列表、正文、标题等信息；最后介绍一下 HTML 的语言形式，首先在浏览器在运行过程中，是不会将这些代码语言等在页面上显示的，另外在开发者编写的过程中，HTML 元素均由一对尖括号包围而成，尖括号内部一般填写对应的符号，代表不同的标签含义。

CSS(Cascading Style Sheets)，中文全称为“层叠样式表”。CSS 是一门计算机语言，目前是由 W3C 组织定义和维护的。它的主要功能是添加字体、间距和颜色等样式至于结构化文档，比如 HTML 文档——具体而言就是会在 HTML 开发的网页基础上，使得交互页面更加美观，起到提升用户交互体验的功能。

JS(JavaScript)，是一种解释型、即时编译型的 Web 编程语言。首先，JavaScript 支持命令式、面向对象以及函数式的编程风格；另外，它的主要作用是用来控制 DOM 元素组件以及页面上的各种事件触发——具体而言就是会在 HTML、CSS 开发的网页基础上，控制与用户与界面进行动态交互的后续响应动作，是用户交

互体验的核心。

JQuery 是一个跨浏览器的库，编写的编程语言为 JavaScript，其主要用途为简化前端编程语言之间交互的流程。首先，jQuery 拥有独特的语法设计，使得在前端中许多操作变得容易开发上手——比如处理事件、选择 DOM 元素、操作文档对象、创建动画效果以及开发 Ajax 程序等开发常用操作；同时，jQuery 库也将创建插件的功能提供了给前端开发人员，这使开发人员可以抽象化高级效果、高级主题化的组件；最后，它对动态网页的功能灵活性的以及基于 web 的应用程序也提供了强大的模块支持。

2.2 后端开发技术基础

后端开发（Back-end development）技术涉及到 Flask、MySQL。本节将会对提到的这些技术进行一一介绍。

Flask 是一款轻量级 web 后端框架，它的编写语言为 Python 语言。它主要适用于微小 Web 后端项目构建的领域，通常以精炼简洁，敏捷轻量著称。首先，在本课题研究中，会通过使用 Flask 框架进行实验研究，尤其在几十行代码编写后，Flask 即可快速部署一个 web 服务，以此来验证课题工作中的一些假设；同时，Flask 是一个入门及其友好的 web 后端框架，拥有详细官方教程以及大量的技术参考文档，对于后端开发者而言，这些是非常好的学习资源，能保证快速上手以及本课题的正常开展；最后，Flask 还提供提供轻量级的 admin 管理后台系统、缓存系统、数据库迁移功能、用户权限管理系统等各种功能，这些额外提供的功能，对新手无疑是非常友好的，也有助于本课题研究的正常推进工作。

MySQL 是目前比较流行的关系型数据库管理系统，并且在 web 端开发这块来讲，MySQL 是最好的关系数据库管理系统软件之一。首先，MySQL 会为课题所需要的数据建立不同的表，并将其分别保存在各自的表中——这样的存储方式与将所有数据放在一个巨型文件内相比，提高了访问数据的速度以及查询数据的灵活性；其次，MySQL 使用标准的 SQL 语言来进行数据表的增删查改，没有太多额外的学习成本；同时，MySQL 可以运行于多个系统上，如 Windows、Centos、Debian 等，也支持多种编程语言，包括 C、C++、Python、Java 等，对于本课题研究来说，各个方面都具有良好的兼容性，因此在本课题中选择了 MySQL 做数据库管理的应用软件。

2.3 词嵌入表示技术

在自然语言处理（NLP）领域中，通常会对中文字符使用词嵌入的技术来进行文本表示，主要做法是将中文字符映射到一个巨大的向量空间中，以一个数学向量为载体，表达该中文字符的文本信息，这也称之为“词嵌入向量”。早期通常会将中文字符进行离散化操作，这样的词嵌入表示方法称之为 one-hot 向量。之后在信息检索领域发展出了词袋模型，这类模型会将文本视为可以用一个装着这些词的袋子来表示，这种表示方式的核心思想是不考虑词的语法关系以及词之间的顺序关系，BM25 模型就是其中之一。随着 NLP 技术的不断发展，又有学者提出了多种静态词向量的表示方法，比如谷歌公司提出来的 Word2Vec^[6]方法。近几年，伴随着预训练思想的横空出世，一系列基于预训练方法得到的动态文本词向量方法百花齐放，而 BERT 模型就是最著名的预训练模型之一，促使了词嵌入文本表示技术发展到了新的一个阶段。

1) One-hot

One-hot 编码，中文名称为“独热编码”，是一种经典的离散化编码方式，主要做法是将分类变量作为由零和一两种数字编码的向量表示——具体而言，首先要求将分类文本字符映射到整数值；然后，将每个整数值表示为二进制向量，将所有的位都标记为 0 值，然后再找到类别对应的索引位置，将其位置标记为 1。以性别特征为例，当一个样本为[“男”]的时候，它的文本独热编码为[1, 0]，而为[“女”]的时候，它的文本独热编码为[0, 1]。这种编码的好处是将文本字符离散特征变换到了欧式空间上，而这些离散特征就会变成某个坐标点，这种做法会让原特征之间的距离计算更具有可解释性。

在本课题中，尝试过对医疗垂类领域的中文文本语料使用 One-hot 编码，但由于此类文本的数据特点为词汇复杂且数量巨大，首先需要利用完整的中文汉语字词库来构建词典，最终得到词汇的数量高达 6 万左右，且随着中文语料库的不断扩张，产生的 One-hot 向量会造成不可想象的维度灾难，这对于目前有限的空间存储资源来说是不可接受的，同时这种编码会有许多语义信息损失，这对于课题预期的最终效果来讲也是不能接受的。

2) BM25

BM25，英文全称为“Best Matching 25”，其中 BM 是 Best Matching 最佳匹配的缩写，25 指的是第 25 次算法迭代。在信息索引领域，BM25 仍然是目前最主流的计算中文文本相似度得分的一个词袋算法模型，其主要做法可以拆解为两个部分——首先，BM25 会考虑到中文文本里每一个词在整体文档里的重要性，

这部分是由 IDF（Inverse Document Frequency，逆文档频率）改进而来的；其次，会考虑到中文文本里每一个词在当前文本的一个重要性，这部分是由 TF（Term Frequency，词频）改进而来的，并且还考虑到了文本本身长度对词频的影响。整合这两个部分之后就可以使用离散化的 BM25 向量来作为中文文本的词嵌入表示，具体计算过程如公式 2-1 和公式 2-2 所示。另外，这种嵌入向量的可解释性与可用性相对于 One-hot 独热编码来说是更加合理的。

$$\text{score}(D, Q) = \sum_{i=1}^n \text{IDF}(q_i) \cdot \frac{f(q_i, D) \cdot (k_1 + 1)}{f(q_i, D) + k_1 \cdot \left(1 - b + b \cdot \frac{|D|}{\text{avgdl}}\right)} \quad (2-1)$$

$$\text{IDF}(q_i) = \ln \left(\frac{N - n(q_i) + 0.5}{n(q_i) + 0.5} + 1 \right) \quad (2-2)$$

本课题尝试使用 BM25 词袋模型对整个中文医疗文本进行了编码处理，这种编码从效果上来讲仍然存在对垂直领域的文本语料适用性一般，且存在一定的 OOV（Out of Vocabulary）问题，但是整体上对于课题的最终目标效果比较符合，且考虑到其存储空间资源和运算时间性能也较好，因此可将 BM25 词袋模型算法作为本课题核心算法之一。

3) Word2Vec

Word2Vec，是由谷歌公司提出的一种基于深度学习的产生静态词嵌入向量的语言模型，是文本表征技术的一大里程碑。Word2Vec 提出了两种不同的神经网络模型来训练词向量，一种是 CBOW（Continuous Bag-of-Words，连续词袋模型），其核心思想是利用上下文的词来预测中心词，然后利用极大似然法去做模型训练；另一种是 Skip-gram（跳字模型），其核心思想与 CBOW 相反，它是利用中心词去预测上下文的词，同样也是利用极大似然法去做模型训练。Word2Vec 的模型训练结构如图 2-1 所示，滑动窗口的大小设置为 2，其中 $w(t)$ 表示中心词的文本表示向量，其余则表示围绕中心词的上下文词的文本表示向量。

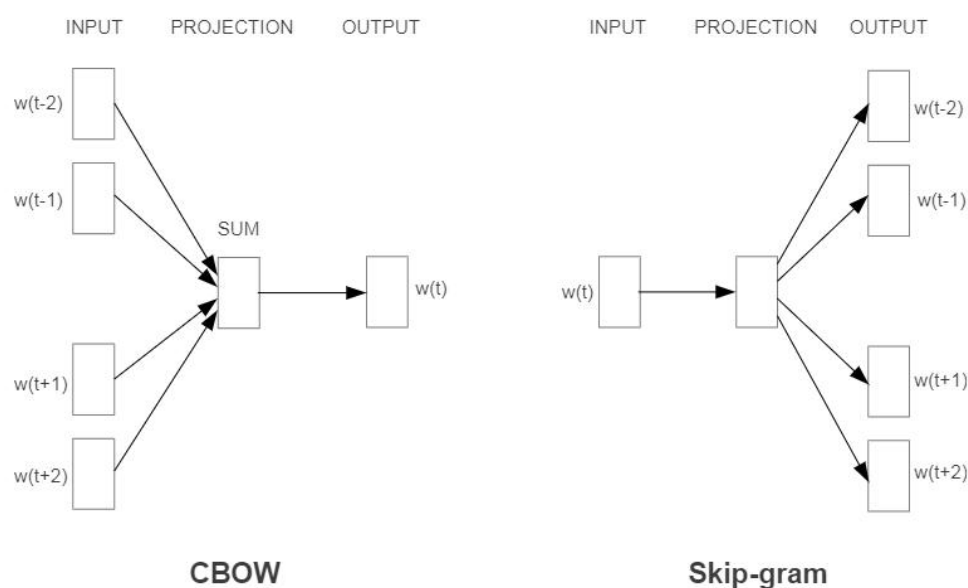


图 2-1 Word2Vec 模型训练结构图

对于本课题而言，Word2Vec 相对于传统的词袋模型或者独热编码的文本表征能力更强，在一定程度上可以考虑到中文词之间的上下文关系，但是在医疗垂类领域下，无法针对本课题的具体任务进行单独优化，并且表现情况与课题预期目标效果还有一定的差距，在实际使用上不太符合课题研究要求。

4) BERT

BERT (Bidirectional Encoder Representations from Transformers)，中文全称为“基于变换器的双向编码器表示技术”，该预训练模型由谷歌公司在 2018 年提出，是 NLP 领域发展的一个重大里程碑，在自然语言理解的多项任务中取得优秀的成绩，成为了目前自然语言处理领域实验中无处不在的基线。BERT 预训练模型的结构如图 2-2 所示，若以中文为基本训练语言，只需要将中文文本以“字粒度”输入至 BERT 模型中，就可以得到整个文本的动态嵌入向量表示。其核心结构是利用 Transformer^[7]模型的多个 Encoder 堆叠而成，通过注意力机制对文本进行建模，具体采用的是多头缩放点积注意力结构(muti-head attention & scale dot product)，注意力具体计算过程如公式 2-3 所示。其中 Q 代表查询向量， K 代表键向量， V 代表值向量， d_k 代表查询向量的维度，该公式通过 SoftMax 函数计算得出一个注意力权重分布矩阵，通过加权求和后得出新的词的动态语义向量表示。

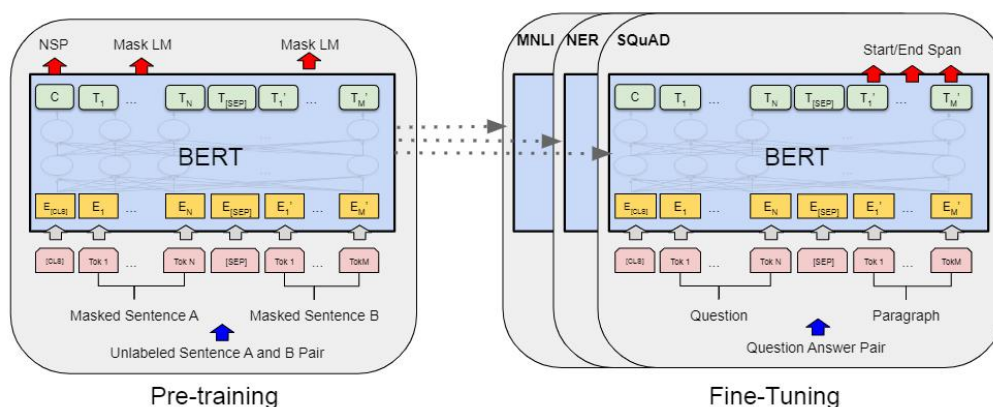


图 2-2 BERT 预训练模型结构图

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (2-3)$$

这种训练方法，首先能够有效解决文本中的长距离信息依赖问题；另外，还可以根据上下文不同，表层词语的多语义性——例如“苹果手机”与“苹果树”里的“苹果”，在 Word2Vec 的静态向量表征中是无法区分的，但在 BERT 里是可以通过动态向量表征来很好的解决这种一词多义的问题的。

对于本课题而言，首先由于 BERT 的结构设计使得课题对于文本的处理成本较低，一定程度上提高了课题研究的可行性；另外，BERT 还能够根据不同领域的语料进行专门的优化，这与课题目标高度吻合；此外，BERT 的初步表现情况比较能满足课题的预期，因此在课题实际使用中，可将 BERT 预训练模型算法作为本课题核心算法之一。

2.4 智能问答系统相关方法

通常智能问答系统的主流实现均为基于深度学习、神经网络的方法，目前按照“使用技术方向”可以分成三类，分别是基于检索式的智能问答系统、基于知识图谱的智能问答系统以及基于机器阅读理解的智能问答系统，具体介绍如下：

其一，基于检索式的智能问答系统：

这类问答系统通常称之为 FAQ（Frequently Asked Questions），中文直译为“常见问题解答”。它的主要做法是通过在已有的“问题-回答”对的数据集合中，找到与用户提问最为相似的问句，并且用户会收到该问句对应的回答作为最终结

果。这类问答系统的优点在于回答的覆盖率较高，同时对于未收录的问题也能靠字面语义匹配取得不错的效果；但是也有比较明显的缺点，比如排序后为第一的答案不一定与用户原问题匹配，这会导致返回结果效果受损，而且对于训练语料的要求也较高。

其二，基于知识图谱的智能问答系统：

这类问答系统通常称之为 KBQA (Knowledge-Based Question Answering)，中文直译为“知识库问答”。它的主要做法是通过构建知识库，并从知识库中搜索与问题相关的实体、关系或属性作为答案返回给用户。这类问答系统的优点在于可以回答知识层面的推理类问题，能够为问题的语义理解提供丰富的背景知识，同时回答准确率较高；但是也有比较明显的缺点，就是构建知识图谱的成本比较高，需要大量人工去标注数据或者高质量的数据集，同时限定知识图谱中必须存在和问题匹配的内容，否则回答不了。

其三，基于机器阅读理解的智能问答系统：

这类问答系统通常称之为 MRC (Machine Reading Comprehension)，中文直译为“机器阅读理解”。它的主要做法是根据给定的上下文，去推断文中哪段话才是问题的答案，例如中高考中的英文阅读理解题目。这类问答系统的优点在于它的回答文本获取容易，不需要进行额外的文本结构化，因此大大减少了对文本数据做结构化处理的时间和人力成本；但是也有比较明显的缺点，就是这类系统需要大量人工标注的数据，以及一个强力的文本理解模型，同时限定了答案必须在文本中出现的条件，并强调必须是文本中的连续片段。

考虑到课题研究的可行性与课题目标的匹配程度，“基于检索式的智能问答系统”的方向作为课题研究的技术基础最能够满足课题的预期，因此可将“基于检索式的智能问答系统”作为本课题的核心算法技术方向之一。

2.5 本章小结

本章主要讲解了基于 BERT 和 BM25 的智能问答系统设计与实现中涉及到的相关理论与技术基础。首先介绍了前端开发技术基础，其次是后端开发技术基础，另外也对词嵌入表示技术的相关理论进行了详细的阐述，最后在智能问答系统在实现层面的相关方法进行了展开。综上所述，以上所提及的内容奠定了基于 BERT 和 BM25 的智能问答系统设计与实现的技术基础与理论基石。

第三章 基于 BERT 和 BM25 的智能问答系统分析与概要设计

本章将着重对基于 BERT 和 BM25 的智能问答系统的具体分析与概要设计进行较为详细的介绍。具体的，3.1 节将会整体介绍目前课题实施过程中存在的问题与挑战，3.2 节将会进行详细的系统需求分析，主要按照可行性分析、功能性需求分析以及非功能性需求分析的顺序进行详细论述，3.3 节将会进行系统的概要设计，分别从系统的架构设计与系统的模块设计进行详细阐述，3.4 节会对本章进行一个总结。

3.1 系统存在问题与挑战

3.1.1 系统概述

课题任务是要求在医疗垂类业务下，用户通过一个在可交互的 web 端页面，输入一个中文文本作为其问题，并且该智能问答系统能够利用相应算法对该中文问题进行字面解析与语义解析，然后将与用户提问最为相似的问题在已有的“问句-回答”对数据集中找到，并且用户会收到将该问句对应的回答作为最终的结果，同时需要保证该结果能够满足用户的正常问答需求。

3.1.2 存在问题与挑战

通过对课题任务的分析，拆解出了在课题任务执行的过程中，存在的问题与挑战，下面将会从系统功能层面、系统性能层面和系统算法层面三个方面分别进行介绍：

首先，在系统功能层面：

1) 系统数据清洗问题：

根据课题任务，为保证领域问答专业性与严谨性，课题研究选用“Chinese medical dialogue data”公开数据集作为数据基底，来做医疗垂类下的智能问答系统——但是该数据集中由于是用户真实问答文本，会含有诸多特殊中文符号、无意义字词、乱码、文本缺失等脏数据或者脏格式，其直接表现为不可供于智能问答系统的召回算法和排序模型使用。

如何清洗与处理数据，直至召回算法与排序模型可直接使用，同时保证预期输出功能正常，是系统功能相关的一个问题点。

2) 系统交互体验问题:

在前后端交互阶段,会产生一个问题,页面的更新提交方式使用一般情况的HTML表单的话,每一次的表单请求都会发送至服务器,并且在服务器返回响应后前端界面又要被重新渲染,这样在用户交互界面就会产生这样一个过程:提交表单→页面清空→填充显示。

以上过程会导致用户体验非常差,这是系统功能层面上需要改进的一个问题点。

系统在用户提问超过查询的候选文本集合范围时,即使返回结果与问题非常不匹配,也会给出最相似的结果,这点不能向用户友好地进行提示,这是系统功能的一个问题点。

其次,在系统性能层面:

1) 系统数据存储问题:

目前候选文本集合十分大,存在着无法将所有候选文本格式化并全部加载入内存进行非持久化存储的现状。那么如何对巨大的候选文本数据集设计一个高效的索引存储结构,来进行合理的持久化存储或非持久化存储,进而保证在智能问答系统算法的召回阶段和排序阶段对于候选文本的正常读写功能,这是系统性能层面的一个问题点。

2) 系统响应时效问题:

在召回阶段,需要将所有候选文本集合与用户在问答系统交互页面上真实输入的中文文本进行 BM25 Ranking,如何高效的对可能是目标文本的集合进行 Ranking 是系统性能层面的一个问题点。

在排序阶段,考虑到 BERT 模型参数量巨大,在预测阶段的耗时是所有模块中最高的,如何能够在正常的用户容忍时间内,对召回服务返回的所有候选进行排序,并保证问答系统的可用性,是系统性能的一个问题点。

最后,在系统算法层面:

1) 系统算法目标定义问题:

系统算法的目标是找到与用户输入的中文提问文本最相似的候选问句文本。因此,如何定义两条不同中文文本的语义相似度,才能更好地适配课题任务目标,达到课题研究是期望效果,是在系统算法层面一个问题点。

2) 系统算法训练优化问题:

在排序阶段,需要对 BERT 预训练模型在利用医疗垂类领域进行微调,以此到达针对课题任务进行专门优化的目的。但是在微调的过程中,所占的内存大小资源比较大,如何保证模型在训练过程中能够快速收敛并且一定程度上达到预期

的训练效果，这是在系统算法层面的一个问题点。

3.2 可行性分析与需求分析

3.2.1 可行性分析

本节将会从技术可行性、经济可行性、社会可行性三个部分按照顺序分别进行分析，来解释说明本智能问答系统的设计与实现的可行性。

1) 技术可行性分析：

首先，本智能问答系统整体采用的是 B/S 架构开发，用户会直接通过浏览器在页面完成交互。其次，在 Web 端开发方面，前端 web 开发使用的 HTML、CSS、JavaScript、JQuery 进行技术实现，Python 语言编写的 Flask Web 框架作为后端 web 开发的技术实现手段。另外，系统功能模块数据存储将交由 MySQL 关系型数据库来完成。同时，算法模型方面选择 Python 语言为主体。最后，随着深度学习与机器学习领域不断的迭代更新，本课题的智能问答系统所涉及算法模型均使用端到端模型，并且利用文本相似度的计算方法实现智能问答的流程。

综上所述，在技术可行性上是符合课题要求的。

2) 经济可行性分析：

在系统 Web 交互页面开发部分，系统所使用的技术均为互联网上公开的开源资料，无须购买额外的软件，因此在开发过程中不会产生太多的经济成本。同时，在智能问答系统算法模型推理训练层面，模型运行和存储需要额外申请一些计算资源，也就是说需要一定内存的服务器与 GPU 算力课题项目才可正常推进，针对这部分的经济成本，由本人顶岗实习所在公司单位承担，且该承担的成本目前处于公司可接受范围之内。

综上所述，在经济可行性上是符合课题要求的。

3) 社会可行性分析：

在国家经历了新冠疫情的大型公共卫生事件后，人们对个人及其亲属的医疗健康问题关注度相比事件发生之前变得非常的高，各个地方的人们都会对医疗保健方面有着个性化的需求。同时，随着医疗行业数字化发展，相关的智能问答系统也将会成为人们日常生活中的健康小帮手，辅助人们对健康的知识了解，有助于医生患者间的信息沟通以及医患信任的建立。

综上所述，在社会可行性上是符合课题要求的。

3.2.2 需求分析

本节将会从功能性需求分析与非功能性需求分析两个方面进行描述，用于确定该系统需要实现哪些功能以及完成哪些工作。

1) 功能性需求分析：

本系统的功能性需求是系统的核心，下面将会从智能问答功能需求、数据处理功能需求、用户交互功能需求三个方面分别进行分析：

其一，智能问答功能需求：智能问答部分的主要功能是根据用户手动输入的中文提问文本，提供在候选库中的相似“问句-答案”的检索服务。功能上需要分为召回层和排序层，分别负责对用户的提问进行字面解析以及语义解析，保证问答的效果。

其二，数据处理功能需求：数据处理部分的主要功能是将系统所需的语料处理成训练集与测试集供给模型训练与评估，同时将语料拆分设计成合适的索引结构来供高效查询，最后对开源的高质量医疗问答数据集的数据处理结果进行持久化或非持久化的存储，供给问答侧所需的格式化数据。

其三，用户交互功能需求：用户交互部分的主要功能是提供给用户一个良好的交互界面，并满足基础的交互功能。具体来讲，首先通过需求确定的方法与课题需求描述，来确定完整的系统需求。一个问答系统需要包含如下功能：一个用户可以创建问答聊天窗口；一个用户可以删除一个聊天窗口；一个用户可以在一个聊天窗口内发出多条消息；问答系统可以分别为一条询问消息返回一条回答结果。

综上，根据系统的用户交互功能需求画出对应的用例图，如图 3-1 所示：

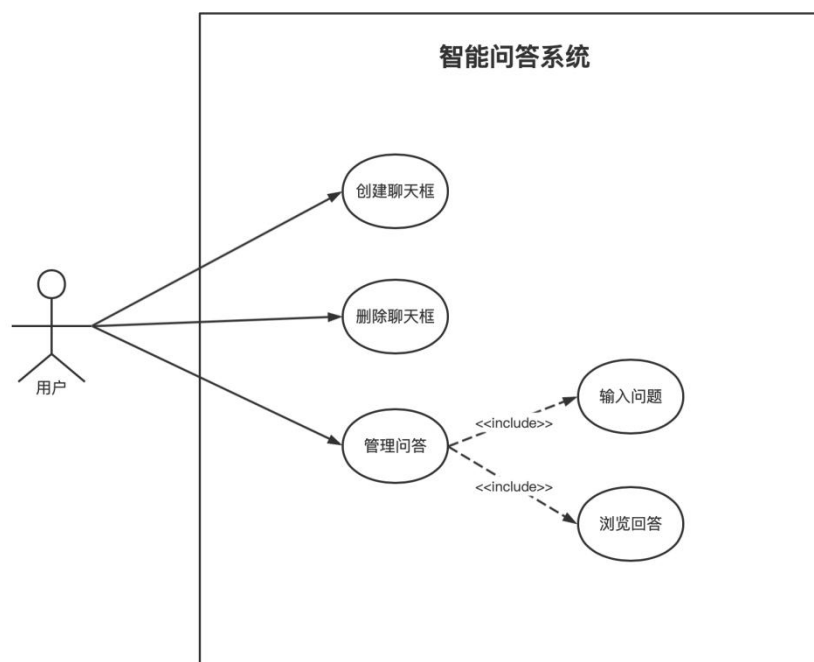


图 3-1 智能问答系统用例图

2) 非功能性需求分析:

在应用软件系统规模越做越大越复杂的时候，系统的非功能性需求就越是不可忽视的，而针对于本系统来说，接下来将会从易用性、性能、可靠性、可维护性四个部分按顺序，分别进行非功能需求分析：

其一，易用性：易用性要求系统的设计需要贴切用户正常的习惯与需求。具体的，易用性需要在使用系统的过程中希望用户不会产生明显的阻碍，因此本系统需要一个简约明朗的一个界面设计，并且在系统出错的时候给予用户一定的提示。

其二，性能：本系统主要提供的是有关中文问答的检索服务，因此对服务的检索速度有一定的要求，需要保证系统从文本解释到检索答案的响应时长保持在 2s 之内。

其三，可靠性：可靠性指产品在规定的限制条件和时间内，不出错地完成规定任务的概率。对于本系统而言，需要保护系统免于承受故障发生和不确定性所造成的后果，因此需要系统有一定的容错性，在服务发生故障时能够快速定位到问题并进行重启。

其四，可维护性：可维护性要求系统在出现错误或故障时能够及时排除关键问题，使得使产品可以正常运作。因此在软件设计应在系统构架上考虑能以尽量少的代价对模块进行解耦，同时打印系统运行日志以及撰写开发文档，保证系统

可维护。

3.3 系统概要设计

本节将从系统架构设计与系统模块设计两方面来介绍系统的概要设计：

1) 系统架构设计：

系统整体采用 B/S 结构，用户交互界面实现交由浏览器负责，业务逻辑实现则交由 web 后端服务器和算法服务器负责，数据存储方面则交由数据库服务器负责，三者构成了 web 应用的三层基础结构。

系统开发方面采用 MVC(Model-View-Controller)架构，模型负责进行数据的表示，视图负责数据的相似，控制器负责对用户的输入进行解释与转发，以及输入的处理步骤，达到控制模型和视图的目的。同时，系统也采用了前后端分离技术——其中，前端使用 HTML、CSS、JQuery 等技术实现页面动态交互与静态页面渲染功能，后端方向的业务处理部分由 Flask 技术框架实现，而智能问答算法的召回层与排序层将使用 Python 语言实现。系统的各模块之间通过各自对外提供的接口进行模块通信，并通过格式化数据后交换信息，这种做法有效提高了模块自身的内聚性，降低了各个模块之间的耦合性。

针对系统的逻辑体系结构进行分析设计，智能问答系统一共包括四种构件，分别是数据库系统构件，对外暴露数据查询接口；聊天窗构件，对外暴露聊天窗接口；模型算法系统构件，对外暴露模型算法接口；还有 Web 系统构件，用于处理业务逻辑。综上所述，画出了对应的系统构件图，如图 3-2 所示

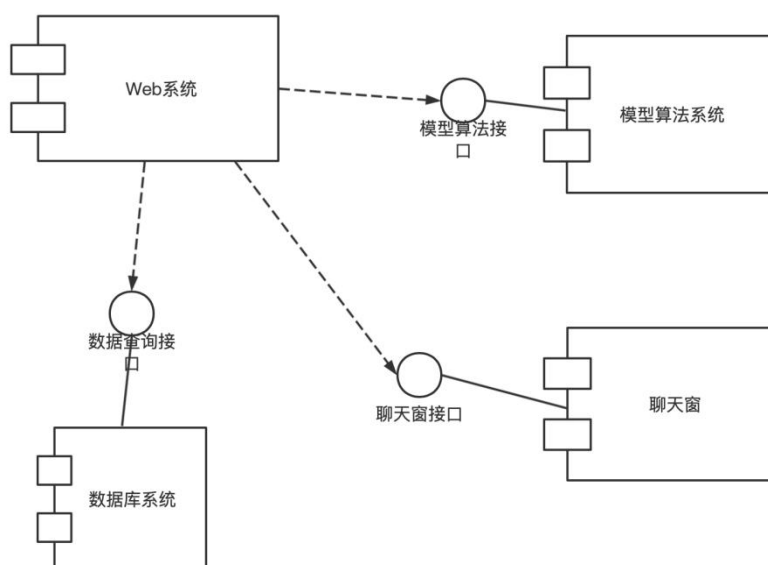


图 3-2 智能问答系统构件图

针对系统的物理体系结构进行分析设计，智能问答系统共包括 4 种节点，分别是负责数据的存储的数据库服务器节点；用于处理系统的业务逻辑的 Web 服务器节点；负责核心算法功能实现的模型算法服务器节点；负责让用户通过浏览器发起问答的用户个人 PC 机节点。综上所述，画出了对应的系统部署图，如图 3-3 所示

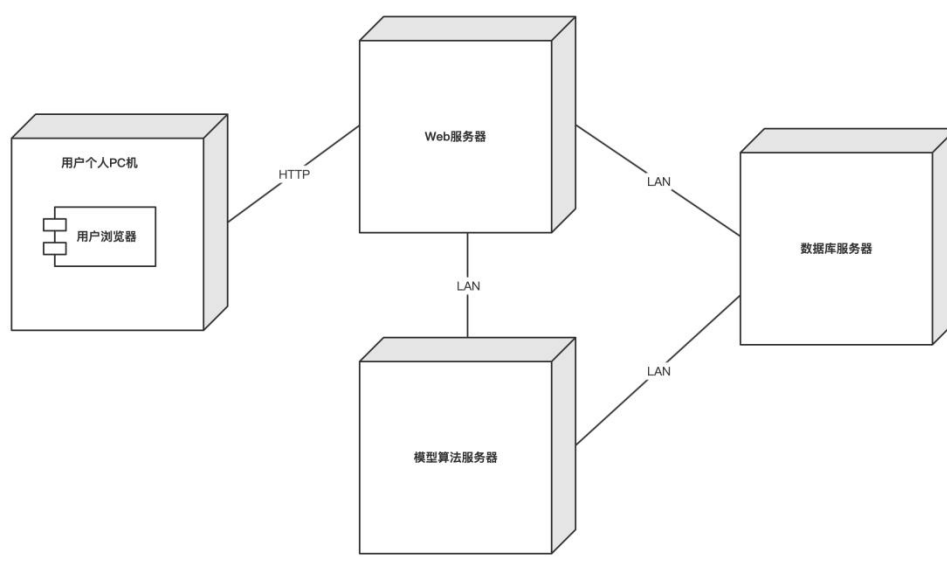


图 3-3 智能问答系统部署图

2) 系统模块设计：

基于上文在需求分析对系统的三大核心功能的分析结果，当前系统可以大致分为智能问答模块、数据处理模块和用户交互模块三个方面。其中，智能问答模块主要负责问句字面检索召回、问句语义解析排序和问句答案组合检索三个方面；数据处理模块主要负责数据清洗和数据存储两个工作；用户交互模块主要负责聊天窗口管理、问答管理两个工作。综上所述，画出了对应的系统模块设计图，如图 3-4：

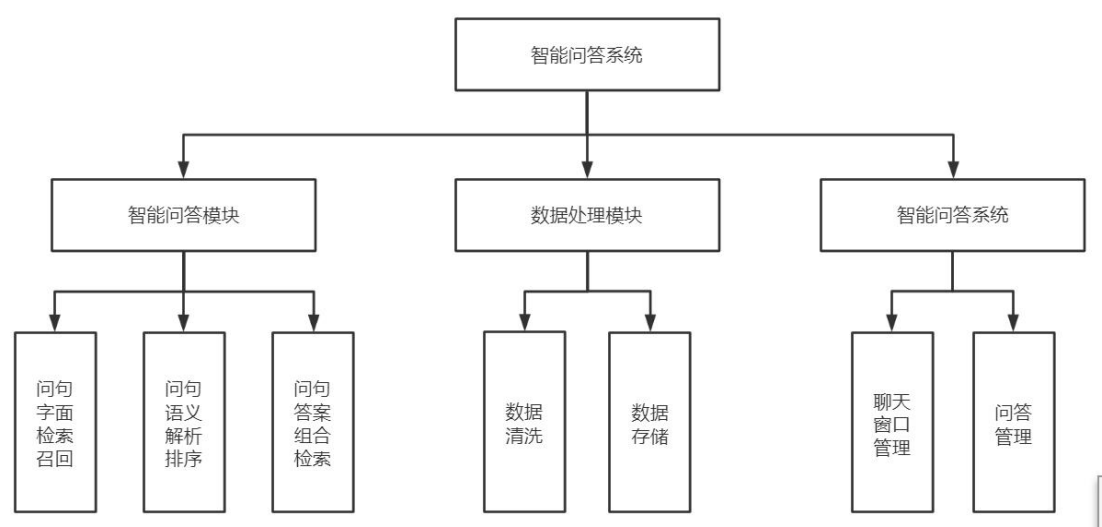


图 3-4 智能问答系统模块设计图

3.4 本章小结

本章主要介绍了基于 BERT 和 BM25 的智能问答系统的具体设计过程。首先整体在目前课题实施过程中存在的问题与挑战进行了介绍，然后对系统的可行性以及需求方面进行了全方位的分析，总结出了本系统的用例图，并以此提出了系统的概要设计，描述了系统的可靠灵活架构设计以及功能健全的系统模块设计，并整理了系统的构件图、部署图以及模块结构图，描述了本系统三大模块的整体设计思想。

第四章 基于 BERT 和 BM25 的智能问答系统详细设计与实现

本章将着重对基于 BERT 和 BM25 的智能问答系统的详细设计与实现过程进行较为详细的介绍。具体的，4.1 节将介绍系统的详细设计，包括各个模块流程的详细设计和基于 BERT 和 BM25 的智能问答系统算法方法详细设计，这两大部分设计是系统实现的重要基石，4.2 节将会介绍系统所在的运行环境，4.3 节将会介绍各个模块具体系统的功能实现，4.4 节将会对本章内容进行总结。

4.1 系统详细设计

4.1.1 系统模块详细设计

4.1.1.1 智能问答模块详细设计

智能问答模块是本系统最核心的模块，其主要内容是根据根据用户问题检索出最相似的问句及其回答，本节将详细介绍智能问答模块的详细设计与流程，其设计到的核心算法方法将会在后文进行介绍。

整个智能问答模块的流程设计如图 4-1 所示。具体流程是：用户输入问题至系统，系统首先会对用户输入的原生文本进行一个文本预处理，然后使用 BM25 算法进行字面解析，并从候选文本中检索召回可能相似的问句，接着使用 BERT 模型对召回层的结果进行语义解析，得出语义层面最相似的问句，然后判别排序结果是否为空，若排序结果为空，则直接将固定的提示语返回给用户界面展示；若排序结果不为空，则在数据库中进行问句-答案对的查询，最后把答案返回给用户界面进行展示。

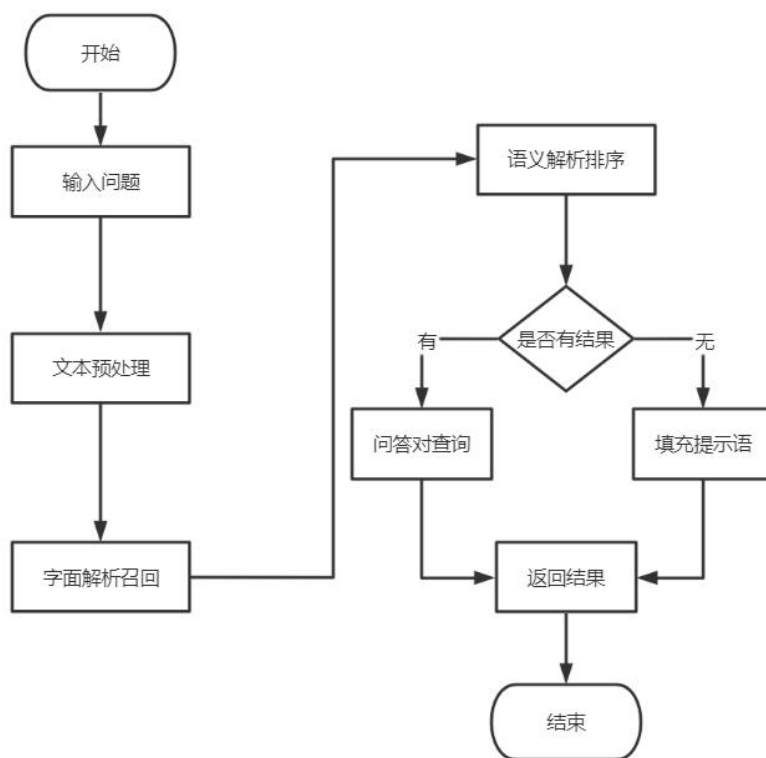


图 4-1 智能问答模块流程图

根据智能问答系统需求分析和体系结构,并从用户需求和用例场景中发现类,可得智能问答模块中的主要类有:问答控制器类(QAController)、问答服务类(QAService)、文本召回服务类(QARecallService)、文本排序服务类(QARankService)、HTTP 工具类(HttpUtil)、问答对查询类(QAStorage)和问答文本信息类(QAInfo)。

首先,智能问答模块反映系统静态结构设计的类图如图 4-2 所示。QAController 是整个模块的入口,主要通过调用 QAService 获取用户问句对于的返回结果。其中, QAService 是实现完整的智能问答流程的类,它会调用 HTTP 工具类进行 URL 解析,然后按顺序调用文本召回服务类、文本排序服务类和问答对查询类,从而返回与用户问句最相似的回答结果。

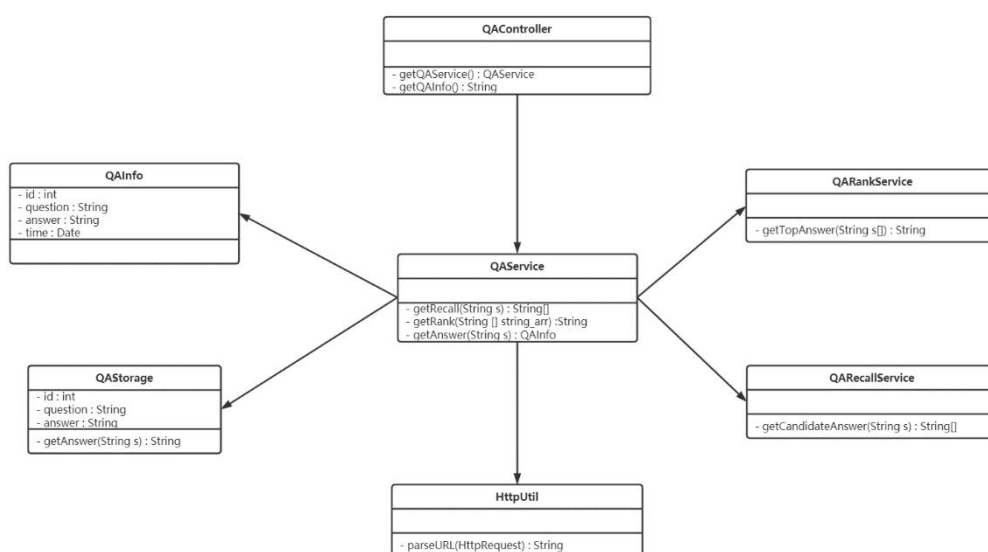


图 4-2 智能问答模块类图

另外，智能问答模块反映系统动态结构设计的时序图如图 4-3 所示。具体的，用户首先会在页面输入问题，浏览器端会向服务端发起一次问答请求，接着服务端会处理完整个智能问答流程，最终将返回结果放置用户界面进行展示。

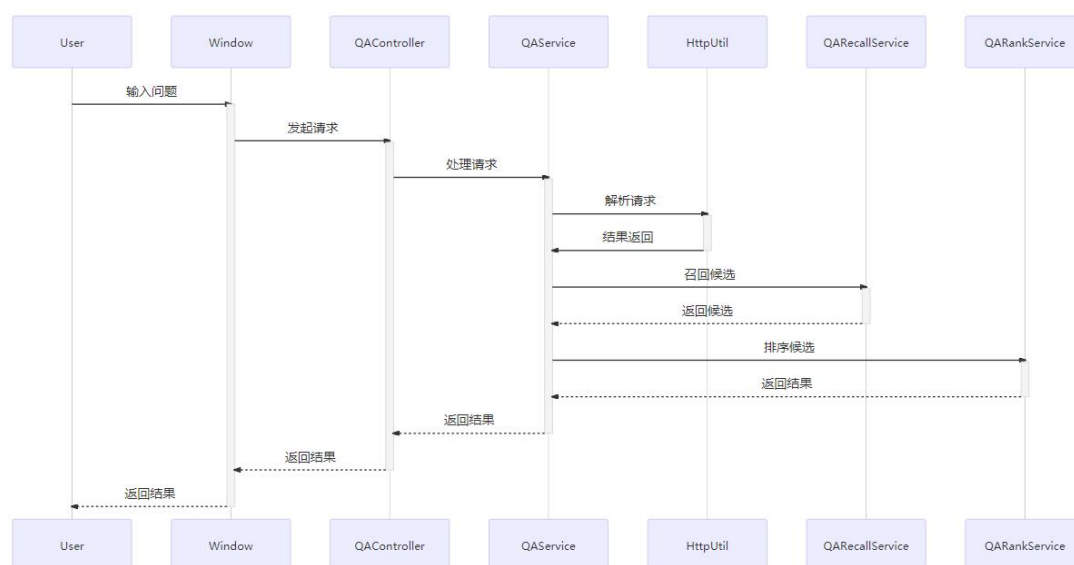


图 4-3 智能问答模块时序图

4.1.1.2 数据处理模块设计

数据处理模块的主要内容是将医疗数据集进行数据清洗与数据存储，本节将详细介绍数据处理模块的具体设计与流程。

数据处理的流程设计，如图 4-4 所示。具体的：首先读取数据集所在文件，然后解析数据格式，若解析过程中发生异常，则打印对应的数据并跳过，否则保留数据至本地文件。接着进行数据清洗，对于有缺失以及乱码的文本进行丢弃，去掉停用词和特殊符号，保留“你、我、他”等一些在此场景中有词间区分度的代词，然后整体进行去重，保持数据的唯一性。同时，将清洗好的干净数据分别进行模型数据集划分、索引构建。最后按照实体属性对的方式，将上述数据持久化存储在对应的数据库中，以上就是系统数据处理的流程设计。

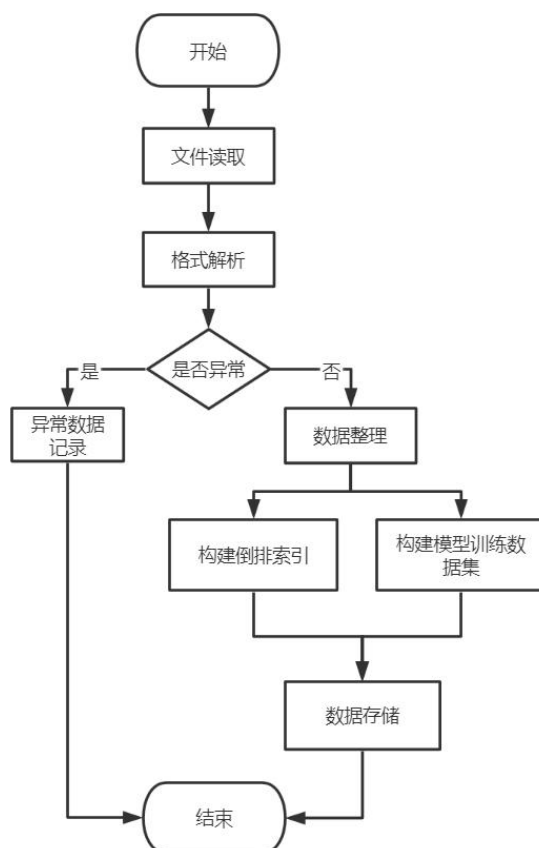


图 4-4 数据处理模块流程图

4.1.1.3 用户交互模块设计

用户交互模块的主要内容是完成用户在使用问答系统的过程中聊天窗口管理、问答管理的操作，本节将详细介绍用户交互模块的具体设计与流程。

用户交互的流程设计如图 4-5 所示。具体流程为：首先用户进行创建聊天框或者选定已有聊天框的操作，若操作为选定已有聊天框，则将对应的历史问答记录加载入界面，否则载入空白界面。然后用户在聊天界面提出问题，最后在前端页面进行答案展示，以上就是系统用户交互的流程设计。

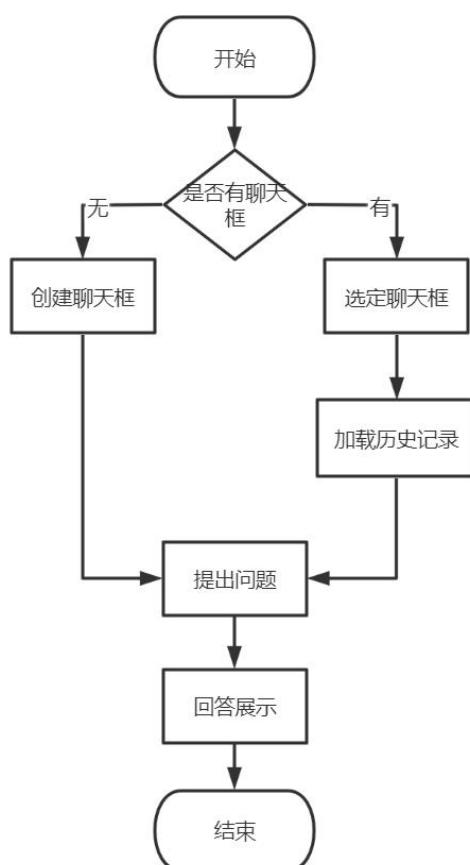


图 4-5 用户交互模块流程图

4.1.2 系统算法详细设计

4.1.2.1 算法提出与应用

本节主要介绍课题研究的算法的提出与应用，主要从算法的提出背景和应用领域及方式进行详细描述。

在用户提出了医疗类问题之后，国内几种常用的搜索引擎通常只会将与问答文本中有一定相关性的网页链接给予用户展示，导致无法直接有效地给出精确的回答，搜索引擎给出的结果对于用户来讲就失去了其权威性、准确性和有效性。

为解决以上问题，同时提升模型的领域专业能力，本文提出了一种基于 BERT 和 BM25 的智能问答算法。详细的算法架构设计图如图 4-6 所示。该算法是一种基于检索式的问答算法，从算法角度看，其拥有着字面解析与语义解析的双重理解优势；从效果角度看，其拥有着对高频问题回答精准的优势。

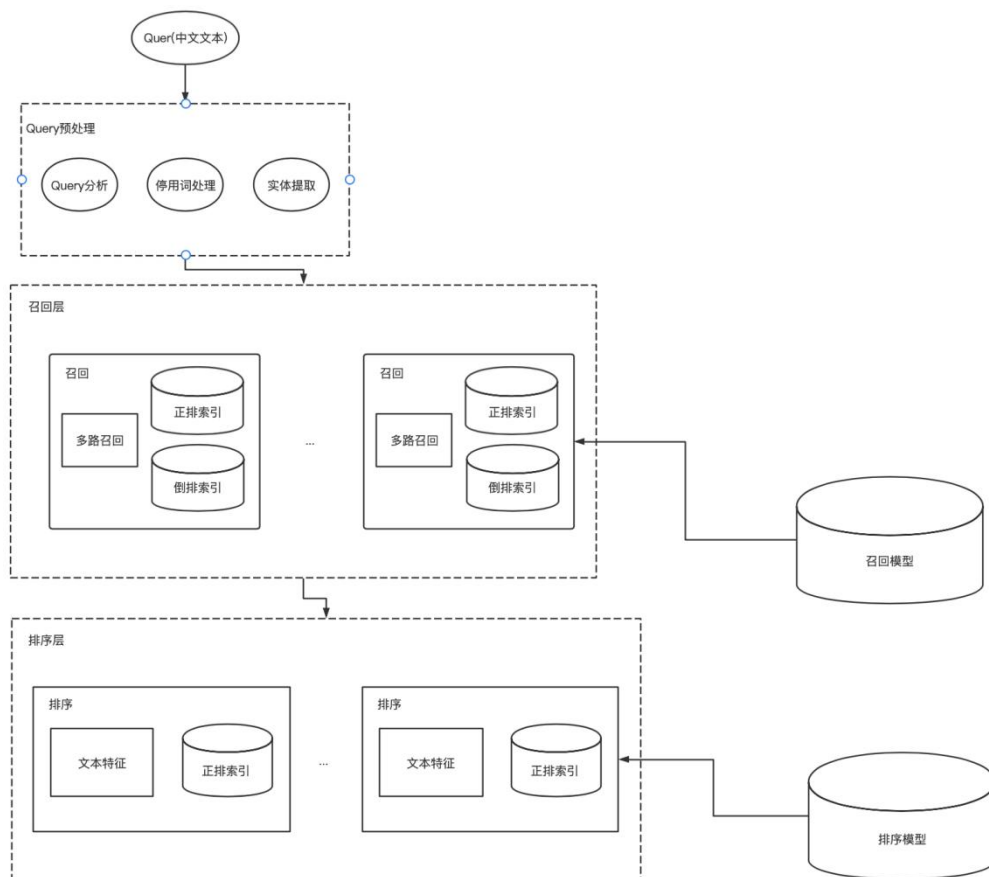


图 4-6 智能问答算法架构设计图

算法执行顺序分为两个步骤：一是字面解析召回算法，二是语义解析排序算法。其中，字面解析召回算法流程如下：根据用户的输入文本，进行初步的分词得到输入文本的 **token** 集合，分别通过不同的 **token** 在倒排索引数据库中取出不同的候选文本，通过 **BM25** 算法分别生成输入文本与候选文本的字面解析后的句向量表示，再利用向量内积来代表召回得分，并获取前一千个得分最高的候选文本集合作为召回层的输出。另外，语义解析排序算法流程如下：将用户的输入文本和召回出来的候选文本集合，通过训练好的 **BERT** 模型分别生成它们对应的语义解析后的句向量表示，再利用文本相似度来计算输入文本与不同候选文本的最终相似得分，并获取最高得分的问句作为结果。最终，将排序算法输出的最相似的问题结果在问句-回答对数据库中检索到对应的答案集。

4.1.2.2 字面解析召回算法

本节主要介绍字面解析基于 **BERT** 和 **BM25** 的智能问答系统中，召回算法的具体算法实现方法与细节。

本节算法的目的是基于 BM25 算法从大量的候选文本中初步筛选出可能相似的候选文本集合，具体字面解析召回算法实现流程与细节如下：

1) 问句分析：

将用户原始的中文输入文本进行问句分析操作。首先，使用中文文本分词工具 jieba^[8]对原始输入文本进行分词操作，得到原生 token 列表。接着，去掉原生 token 列表中的停用词（如“这”、“那”、“怎么”等）和特殊符号（如“*”、“%”等），保留“你、我、他”等一些在此场景中有词间区分度的代词。最后，统计每个 token 在原始中文输入文本中的词频和逆文档频率，利用 BM25 算法进行计算，得到该问句文本的 BM25 向量作为字面词嵌入表示。

2) 索引召回：

以在“问句分析”部分得到的 token 集合为数据基础，首先遍历每个的 token 在倒排索引结构的位置，并根据位置将持久化的数据从本地数据库进行读操作，从中取出候选问句文本拉链，并对拉链进行遍历得到一个个候选问句文本。接着，将所有 token 对应的候选问句文本集合放在一起进行去重，防止重复计算。最后，根据 BM25 算法，计算出每个 token 对应候选文本的 BM25 向量作为字面词嵌入表示。

3) 灵活排序：

由于只需要获取前一千个得分最高的候选文本集合即可满足召回层的目标，因此在召回算法中采用小根堆的数据结构，其特点是堆中某个结点的值总是不小于其父结点的值且堆一定是一棵完全二叉树，这样就避免了候选文本全部排序，有着提升算法灵活性的优势。

具体灵活排序过程如下：对候选文本集合进行遍历，遍历过程中利用向量内积来计算一个候选文本的 BM25 向量与原生输入问句文本的 BM25 向量的相似度得分，然后构建(候选文本，得分)的二元组，接着将该二元组放入以“得分”为排序依据的小根堆里，进行判断：若小根堆的大小超过了一千，则弹出堆顶元素，直至小根堆的大小小于等于一千；否则，不做任何操作。最后，把小根堆的所有二元组取出，并且将二元组集合里的“候选文本”集合作为召回层的最终输出。

4.1.2.3 语义解析排序算法

本节主要介绍字面解析基于 BERT 和 BM25 的智能问答系统中，排序算法的具体算法实现方法与细节。

本节算法的目的是通过语义解析，从少量的候选文本中筛选出最可能相似的候选文本。具体语义解析排序算法将会分为模型训练与模型预测两个方向进行介

绍。

首先，介绍的是模型训练部分——该部分主要讲述了如何基于 BERT 模型进行训练，并使得模型能够达到一定领域任务文本语义理解的目标的过程。具体算法模型训练架构图如图 4-7 所示，整体采用的是端到端（End-to-end）的训练方式，模型的输入是医疗问句文本，接着再通过可微调的 BERT 预训练模型将文本字符串映射成句子表示向量；损失层利用的 Triplet Loss 进行损失目标定义，让模型在已有的医疗语料数据分布上，朝着课题目标进行拟合。

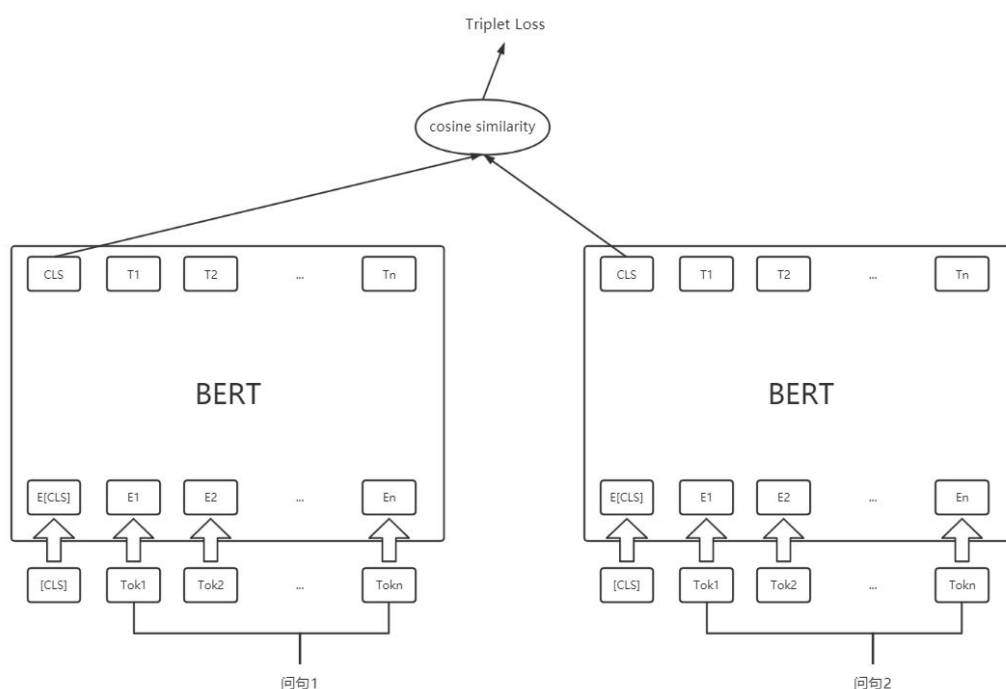


图 4-7 智能问答算法模型训练架构图

具体语义解析排序算法训练实现流程与细节如下：

1) 输入层：

在样本划分层面，将样本按照训练集：验证集：测试集=8：1：1 来进行划分。

在样本构造层面，由于损失层利用的是 Triplet Loss，因此输入将由一个三元组组成，分别是“anchor”、“positive”、“negative”组成。其中“anchor”代表“基准正例”；“positive”代表正例，其文本与“anchor”文本语义相同；“negative”，其文本与“anchor”文本语义不同。

2) 嵌入层：

在本课题研究的任务中，需要根据不同的数据集文本生成合适的词嵌入向量或者句嵌入向量本排序算法会利用微调（fine-tuning）的 BERT 中文预训练模型来

生成语义层面的动态词嵌入向量和句嵌入向量,其中句嵌入向量为 BERT 中[CLS] token 对应的向量,其余细节上文已经提及,此处不再赘述。

3) 损失层:

损失采用 Triplet Loss, 具体计算方式如公式 4-1 所示。Triplet Loss 与在分类预测中常用的 softmax 交叉熵损失函数不同, 它的优势在于更细处上的区分——当两个输入较为相似时, triplet loss 可以学习到输入的更优异的表示, 与交叉熵损失函数的极大似然法不同, triplet loss 能够对两个输入的不同点进行更细致的建模, 这点与课题的目标更为契合。

$$L = \max(d(a, p) - d(a, n) + \text{margin}, 0) \quad (4-1)$$

其中 triplet loss 的距离度量选用向量的余弦相似度, 其计算方式如公式 4-2 所示, 这种距离度量会使得文本向量空间中语义越紧密的文本它们的句向量夹角越小越好。

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n (A_i)^2} \times \sqrt{\sum_{i=1}^n (B_i)^2}} \quad (4-2)$$

同时为了使得不同难度的样本有区分度, 利用 margin 超参数将训练样本区分为简单三元组、一般三元组、困难三元组, 如图 4-8 所示, 期望通过合适的样本划分提高训练效果。

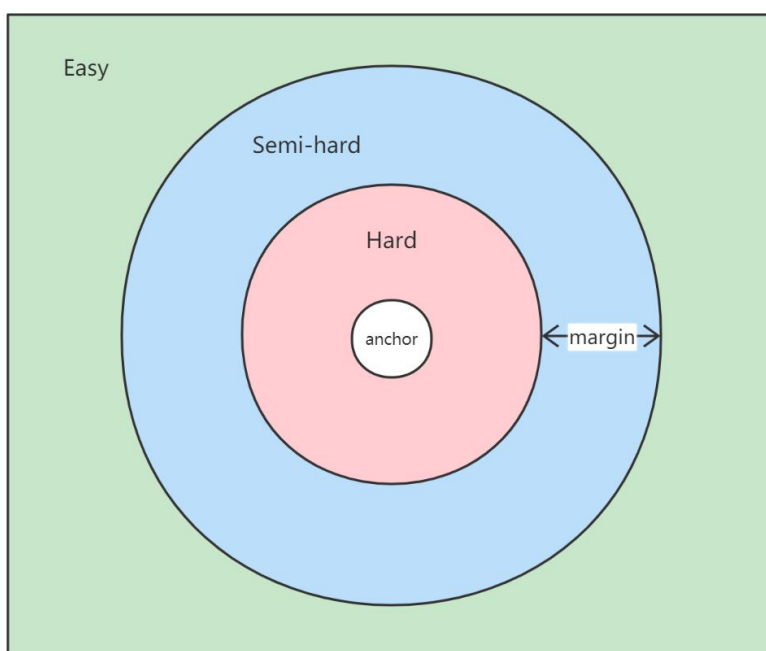


图 4-8 margin 划分样本图

其次，介绍的是模型预测部分——该部分主要内容是如何基于 BERT 模型进行线上预测，并使得模型能够达到一定领域任务文本语义理解并排序的目标的过程。具体排序模型预测介绍将分为嵌入层和预测层两部分，实现流程和细节如下：

1) 嵌入层：

上文已经重点阐述过了模型的训练过程，本排序算法会利用该训练好的 BERT 中文模型，来为候选文本以及原生问句输入文本生成语义解析后的动态词嵌入向量和句嵌入向量，其中句嵌入向量为 BERT 中[CLS] token 对应的向量。

2) 预测层：

求出候选文本句向量和原生问句输入文本句向量直接的距离得分，距离度量选用向量的余弦相似度。然后按照距离得分对候选集合进行排序。最后，返回得分最高的那个候选文本，作为排序层最终的输出。

4.1.2.4 算法实验评价与分析

本节的实验主要是依据“Chinese medical dialogue data 中文医疗对话数据集”进行的构建。其中实验使用数据集共约八十万条，其中八万条中文问答语料用于召回层算法与排序层算法模型的模型评估测试，其余七十二万条用于排序层的神经网络模型训练。具体的，将会分别从字面解析召回算法与语义解析排序算法两个方面进行算法实验评价与分析。

1. 字面解析召回算法实验评价与分析：

字面解析召回使用的是 BM25 词袋算法模型，为了评估该算法模型的召回效果，同时考虑到召回层是对于唯一的一个相关结果进行评估，因此将使用问答类检索常用指标 MRR（Mean Reciprocal Rank）来进行评估，该指标具体含义是指多个查询文本语句的排名倒数的均值。同时为了证明 BM25 的有效性，将与 TF-IDF 模型进行对比，评估结果如表 4-1 所示。

表 4-1 召回算法评估表

模型	MRR
TF-IDF	0.101
BM25	0.143

实验结果表明，使用 BM25 可以提升召回检索能力，说明 BM25 词袋算法模型在一定程度上可以满足当前系统的召回需求。

2. 语义解析排序算法实验评价与分析：

语义解析召回使用的是 BERT 词袋算法模块，为了评估该算法模型的排序效果，将使用准确率（Precision）、召回率（Recall）和 F1 值（F1-score）作为模型的评估指标，评估结果如表 4-2 所示。

表 4-2 排序算法评估表

模型	Precision	Recall	F1-score
BERT	0.73	0.75	0.74
BERT[微调]	0.78	0.82	0.80

实验结果表明，使用医疗语料数据集微调后的 BERT 预训练模型，相比于没有微调的 BERT 模型能够获得更高的精确度，说明微调在一定程度上能够提高模型的准确率与召回率，并且当前训练出来的模型可以满足基本的医疗问答需求。

4.2 运行环境说明

本系统整体采用 B/S 结构，HTML、CSS、JavaScript 等前端技术负责用户交互界面实现，Flask 等后端技术负责 web 后端业务服务实现，Python 等机器学习技术负责系统核心算法服务实现，MySQL 数据库负责数据存储实现。本文提出的基于 BERT 和 BM25 的智能问答系统的运行环境，如表 4-3 所示。

表 4-3 系统运行环境表

参数	值
操作系统	Debian 9.0
内存大小	128G
Python	3.6
Tensorflow	2.7
Flask	2.0.2
MySQL	5.7.17
Chrome 浏览器	100.0.4896.127

4.3 系统功能实现

基于上文提出的基于 BERT 和 BM25 的智能问答系统设计，本节将分模块介绍该系统的具体功能实现。

4.3.1 系统智能问答模块实现

系统智能问答模块围绕着基于 BERT 和 BM25 的智能问答算法展开，是本系统实现的核心部分，下文将对智能问答系统的算法处理实现进行着重介绍，分别从召回层与排序层两个方面阐述。

1) 召回层

智能问答模块的召回层是利用 BM25 算法进行的实现，在课题研究的实施过程中，其实验具体参数设置为： $k_1=2.0$ ， $b=0.75$ ，该参数值均为经验值。在召回时，获取到原始中文文本的 token 词典后，获取预先统计好的每个 token 在原始中文输入文本中的词频和逆文档频率，并利用 BM25 算法进行计算，得到该问句文本的 BM25 向量作为字面词嵌入表示。接着遍历每个的 token 在倒排索引结构的位置，从中取出候选问句文本拉链——过程中会涉及数据库读入操作，该部分由 Flask 框架中提供的数据库连接池进行实现，具体为：在创建数据库连接之后，将连接放在池中并再次使用，这样就不必建立新的连接。相比于为每一次倒排索引库的查询打开和维护数据库连接来讲，数据库连接池的操作十分节省连接时间，提高了智能问答模块召回层在数据库上执行命令的性能。最后计算候选文本 BM25 向量和原生中文文本 BM25 向量的内积作为召回得分，并利用一个小根堆记录召回得分排名在前一千的候选文本，这样就避免了候选文本全部排序，提升了召回层的系统性能，召回层部分编码实现如图 4-9 所示。

```
class RetrievalService():

    def build(self):
        from models import Term, StopWord
        from __init__ import app
        # 1.获取停用词
        raw_stop_words = StopWord.query.all()
        self.stop_words = set()
        for stop_word in raw_stop_words:
            self.stop_words.add(stop_word.stop_word_str)

        app.logger.info("Init stop_words done")

        # 2.获取term iqf
        self.term_iqf_dict = dict()
        raw_terms = Term.query.all()
        self.average_iqf = 0
        for term in raw_terms:
            self.term_iqf_dict[term.term_str] = term.term_iqf
            self.average_iqf += term.term_iqf
        self.average_iqf = self.average_iqf / len(self.term_iqf_dict.keys())

        app.logger.info("Init term_iqf dict done,avg_iqf:", str(round(self.average_iqf, 4)))

        return

    def retrieval(self, source_question: str, top_n=1000):
        """召回和source_question相关的candidate_question,并取top_n送往排序层"""

        # 1.按照正常模式进行分词
        source_token_dict = self.get_token(source_question)
        # 2.从数据库里取,做term召回
        candidate_question_list = self.term_recall(source_token_dict)
        # 3.计算
        retrieval_candidate_question_list = self.retrieval_rank(source_token_dict=source_token_dict,
                                                                candidate_question_list=candidate_question_list,
                                                                top_n=top_n)
```

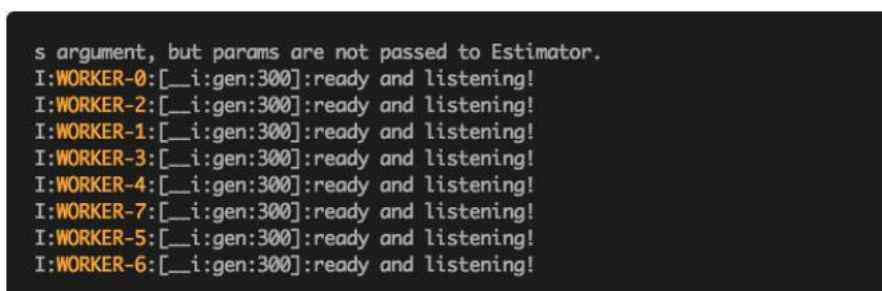
图 4-9 召回层编码实现图

2) 排序层

智能问答模块的排序层是利用 BERT 预训练算法模型进行实现的，在课题研究的实施过程中，将从模型训练实现以及模型部署实现两个方面进行介绍。

首先，是模型训练实现部分的介绍。实验具体训练参数设置为：训练语言为中文，模型层数为 12，隐藏层大小为 768，注意力头数目为 12，模型参数训练设置均为可训练，训练批次大小为 16，学习率设置为 $1e-5$ ，最大句子长度为 128，训练循环 epoch 设置为 3。本系统使用的 BERT 预训练模型包括三个输入，分别为词向量（Token Embedding）、句子类别向量（Segment Embedding）和位置向量（Position Embedding），需要按照向量加法的原则，将三个输入整理成 BERT 预训练模型的最终输入。在损失计算的过程中，取得 BERT 模型最后一层的“[CLS]” token 作为整句话的语义编码向量，然后按照 triplet loss 的计算公式，计算向量间的余弦相似度与损失值，最后按照链式法则进行梯度回传，完成一次训练。

其次，是模型部署实现部分的介绍。模型具体部署架构采用微服务架构，将基于 BERT 模型的排序层看做一类服务，并将其服务化——在一次请求到来时，需要通过负载均衡等手段向智能问答排序层 BERT 模型集群请求而不是单独的 BERT 模型。这部分利用 Python 语言编写的 CLIP-as-service^[9]包进行实现，相比于单独请求一个 BERT 模型的做法，BERT 服务化的做法提升了智能问答模块排序层的整体响应速度，从而使得智能问答系统的总体性能良好，提升系统体验，模型部署效果如图 4-10 所示。



```
s argument, but params are not passed to Estimator.
I:WORKER-0:[_i:gen:300]:ready and listening!
I:WORKER-2:[_i:gen:300]:ready and listening!
I:WORKER-1:[_i:gen:300]:ready and listening!
I:WORKER-3:[_i:gen:300]:ready and listening!
I:WORKER-4:[_i:gen:300]:ready and listening!
I:WORKER-7:[_i:gen:300]:ready and listening!
I:WORKER-5:[_i:gen:300]:ready and listening!
I:WORKER-6:[_i:gen:300]:ready and listening!
```

图 4-10 排序模型部署效果图

4.3.2 系统数据处理模块实现

数据处理模块需要具体实现的主要内容分两个方面：在一方面，是搭建智能问答系统召回层所需的倒排索引库与问答对数据；在另一方面，是构建智能问答系统排序层算法模型所需的训练样本数据。下面将分别介绍系统各个部分的数据

处理实现流程。

1) 倒排索引库与问答对构建实现

为考虑到数据的权威性以及专业性，倒排索引库构建部分的数据源来自“Chinese medical dialogue data”数据集。

倒排索引库的实现过程为：将每个问句进行分词后，结构化为“key-value”存储，存储格式为 json 字符串，其中 key 代表分词后的 token，value 代表问句 sentence，以问句“高血压患者能吃党参吗？”为例子，最终的数据处理效果如图 4-11 所示。

```
{ 'token': '高血压', 'sentence': '高血压患者能吃党参吗?' }  
{ 'token': '患者', 'sentence': '高血压患者能吃党参吗?' }  
{ 'token': '吃', 'sentence': '高血压患者能吃党参吗?' }  
{ 'token': '党参', 'sentence': '高血压患者能吃党参吗?' }
```

图 4-11 倒排索引构建效果展示图

问答对构建的实现过程较为简单，归因于“Chinese medical dialogue data”数据集的数据格式比较吻合。可以直接提取出原数据集中的问句和回答，然后结构化为“key-value”存储，存储格式为 json 字符串，其中 key 代表问句文本 question，value 代表回答文本 answer，以问句“高血压患者能吃党参吗？”为例子，最终的数据处理效果如图 4-12 所示。

```
{ 'question': '高血压患者能吃党参吗?', 'answer': '高血压病人可以口服党参的。党参有降血脂，降血压的作用，可  
{ 'question': '老年人高血压一般如何治疗?', 'answer': '高血压，这是老年人常见的心血管病，血管老化硬化，血压  
{ 'question': '糖尿病还会进行遗传吗?', 'answer': '2型糖尿病的隔代遗传概率为父母患糖尿病，临产的发生率为40
```

图 4-12 问答对构建效果展示图

最后通过 Flask 提供的 ORM 框架技术，编写对应的模板类后，通过模板类将上述处理后的数据写入至 MySQL 数据库进行持久化存储。

2) 智能问答系统算法模型训练样本构建实现

为考虑到训练数据的多样性以及丰富度，智能问答系统算法模型训练样本部分的数据源来自“Chinese medical dialogue data”数据集。将原数据按照不同科室进行整理，得到样本在科室维度上的一个大致分布，其分布柱状图如图 4-13 所示。

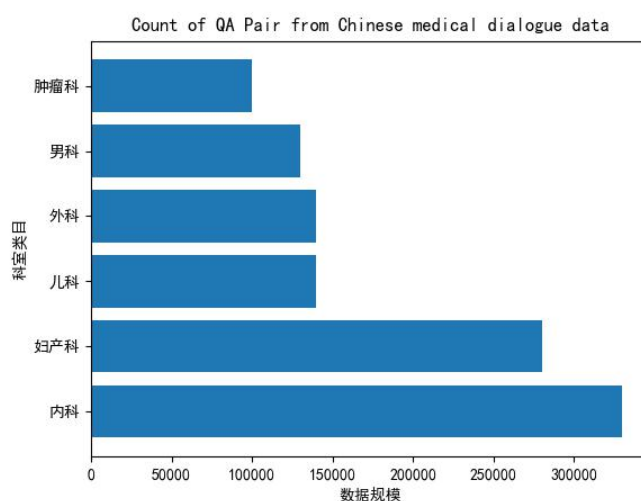


图 4-13 数据集科室分布柱状图

训练样本数据均为结构化数据，每组数据包含三个元素，其组内元素通过制表符分割，分别表示“锚框文本”“正例文本”和“负例文本”，其中“负例文本”的具体构造方法为在整个候选池中按照均匀分布随机抽选一个作为，也称之为“随机负样本”。同时，为了方便模型训练与评估，需要先对数据集进行一个随机打散，接着对处理后的数据集进行划分，划分比例为：训练集：验证集：测试集=8：1：1。

4.3.3 系统用户交互模块实现

用户交互模块的实现主要分为两个方面：一是聊天窗口管理，二是问答管理的交互页面构建。

1) 聊天窗口管理实现

当使用者第一次进入系统交互页面时，系统会自动创建一个新的聊天框，免于用户手动创建聊天框进行问答，减少系统对用户的使用摩擦力。

由于该功能大多仅涉及代码编写逻辑，因此不再对聊天窗口管理部分的其他实现过程进行过多赘述。

2) 问答管理实现

首先，当使用者在用户交互界面输入中文文本并点击发送按钮后，前端会通过 AJAX 方法来请求后台服务，本质上它的实现是在 JavaScript 中使用 XMLHttpRequest 进行 HTTP 的请求，而在实际开发实施的过程中，课题选择使用 JQuery 提供的 AJAX 功能。但同时，引入 AJAX 也带来了无法在页面中向和

域名不一样的服务器发送请求的问题。针对新存在的问题，在本系统问答管理实现中使用了代理的方式作为实际解决方法，具体的，是用在当前页面所在的域的服务端做代理。接着，在后台服务返回回答文本数据时，利用 Flask 框架里支持的 Jinja 模板进行页面渲染完成实现。

同时，为了方便用户的阅读习惯，将使用者的聊天气泡放在聊天框右侧，将智能问答系统的聊天气泡放在聊天框左侧，减少用户对系统的使用摩擦力。

另外，为了方便用户浏览历史问答记录，前端实现过程并未选择内容替换式的实现（即只实现两个固定的聊天气泡，在问答过程中进行内容替换即可），而是选择尾部插入式的实现（即实现多个聊天气泡，在问答过程中按照时间发生顺序，在聊天页面尾部插入聊天气泡），减少用户对系统的使用困惑性，提升系统的产品效用。

4.4 本章小结

本章首先介绍了系统的详细设计，其一介绍了整个智能问答系统的模块详细设计，包括智能问答模块、数据处理模块、用户交互模块三个方面的流程详细设计，其二介绍了智能问答系统的核心算法方法设计，包括了算法的提出与应用、算法方法中的各个模块的算法流程，以及对提出的算法方法进行了实验验证并完成结果分析。接着，介绍了系统所在的运行环境。最后，详细阐述了基于 BERT 和 BM25 的智能问答系统各个模块的具体功能实现。

第五章 基于 BERT 和 BM25 的智能问答系统测试

本章将详细介绍系统三大模块功能测试过程与系统非功能测试过程。具体的，5.1 节将从智能问答系统的功能性测试和非功能性测试两方面按照顺序进行测试流程与测试结果的介绍，5.2 节将会展示系统具体的运行效果，5.3 节将总结一下本章内容。

5.1 系统测试

系统测试是软件生命周期构建中的重要步骤，它担负着确保系统能够按照要求正常运行的重要责任。本节主要的测试方法为黑盒测试，从使用者的角度来测试输入与输出的对应关系是否符合预期。本节将会介绍对基于 BERT 和 BM25 的智能问答系统的功能性需求和非功能性需求进行的测试过程与结果，以此来确保本课题研究“基于 BERT 和 BM25 的智能问答系统”的可靠性与可用性，最终会通过测试用例表的形式来展现。

5.1.1 功能性测试

功能性测试是按功能要求对软件进行的测试，它担负着确保系统功能能够按照要求正常运行的责任。具体的，系统需要通过测试其所拥有的全部特点和功能，来确保符合系统的需求。本节主要从智能问答模块、数据处理模块、用户交互模块三个方面进行模块功能性测试，来以确保基于 BERT 和 BM25 的智能问答系统的功能可以正常运行。

智能问答模块部分功能的测试用例如表 5-1 所示，从表中可以得出，智能问答模块各个部分的功能均可以测试通过。

表 5-1 智能问答模块功能测试用例表

编号	用例名称	前置条件	流程	期望输出	结果
1	字面解析召回	输入框输入问句 文本	请求模型打分 并排序	返回召回结果	通过

2	语义解析排序	召回层返回召回 文本集合	请求模型打分 并排序	返回排序结果	通过
3	问答对检索	排序层返回排序 后列表	取得分最高项 问句进行检索	返回对应的回 答	通过

数据处理模块部分功能的测试用例如表 5-2 所示，从表中可以得出，智能问答数据处理模块各个部分的功能均可以测试通过。

表 5-2 数据处理模块功能测试用例表

编号	用例名称	前置条件	流程	期望输出	结果
1	数据读入	预备初始文本数 据集	格式解析与异 常检测	解析成功	通过
2	数据清洗	数据读入正常	倒排索引和模 型训练数据集 建立与去重复 项	清洗成功	通过
3	数据存储	数据处理完毕	将数据存储至 本地文件与数 据库	存储成功	通过

用户交互模块部分功能的测试用例如表 5-3 所示，主要针对交互功能进行测试，从表中可以得出，用户交互模块各个部分的功能均可以测试通过。

表 5-3 用户交互模块测试用例表

编号	用例名称	前置条件	流程	期望输出	结果
1	发送问题	聊天输入框输入 问句文本	输入问句，点 击发送按钮	发送成功	通过

2	问答显示	完成智能问答	页面按顺序显示问答文本	显示成功	通过
3	查看历史记录	存在历史问答记录	页面按时间顺序显示历史文本	查看成功	通过

5.1.2 非功能性测试

非功能测试用于检查系统的非功能性方面的能力，本节将会对基于 BERT 和 BM25 的智能问答系统的兼容性、性能、可靠性三个方面进行测试过程与结果的介绍。其中，兼容性测试使用的是主流的浏览器进行对比，性能与可靠性采用 Flask 支持的测试框架进行测试。

表 5-4 展示了基于 BERT 和 BM25 的智能问答系统的非功能测试用例，从表中可以得出结论：当前基于 BERT 和 BM25 的智能问答系统性能优异，能够满足用户的正常问答需求。

表 5-4 系统非功能测试用例表

编号	测试用例	测试方案	期望输出	结果
1	系统可靠性	重复向查询接口发送请求，测试进程是否正常存活	系统服务进程正常运行	通过
2	系统并发量	利用多个浏览器页面模拟多个用户同时请求的场景，同时页面访问失败率	系统最大 QPS 不少于 60	通过
3	系统兼容性	测试不同常用浏览器打开页面效果	各个浏览器效果一致	通过

5.2 系统运行效果展示

本节将从用户体验的角度，对本文提出的基于 BERT 和 BM25 的智能问答系统的实际运行效果进行展示，下面将介绍具体展示情况。

首先，系统通过 Flask 部署在本地服务器上，利用浏览器即可访问到智能问答系统的用户交互页面，具体展示效果如图 5-1 所示。

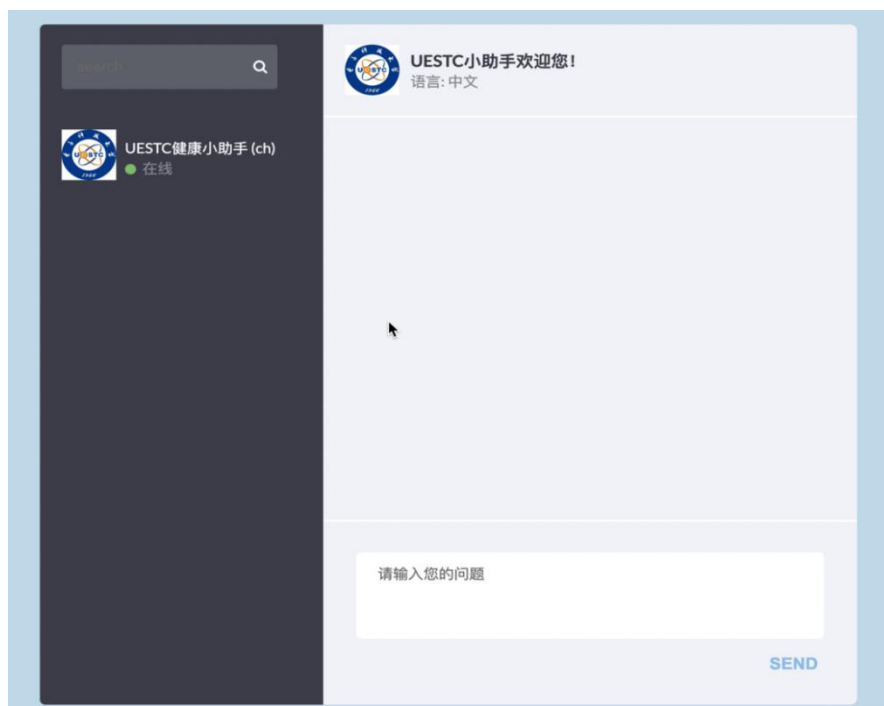


图 5-1 系统交互页面展示图

其次，用户在聊天框输入需要查询的医疗相关的问题，系统就会在数据库中检索，返回对应的回答，例如用户查询问题“得了糖尿病能吃水果吗”，系统将会执行基于 BERT 和 BM25 的智能问答算法，并返回检索库中得分最高的结果，具体展示效果如图 5-2 所示。



图 5-2 系统问答页面展示图

同时，如果用户在聊天框输入了非医疗相关的问题，如“奔驰和宝马谁好”这种汽车领域的问题，系统会进行友好提示，具体展示效果如图 5-3 所示。



图 5-3 系统提示页面展示图

另外，系统支持查看历史问答记录，例如查看曾经询问过的问题“高血压能去做剧烈运动吗”，用户在聊天框向上滑动即可查看历史消息，具体展示效果如图 5-4 所示。



图 5-4 系统历史页面展示图

5.3 本章小结

本章详细介绍了具体的基于 BERT 和 BM25 的智能问答系统的测试过程与运行效果,首先介绍系统所使用的测试方法,然后详细介绍了系统主要的模块功能测试与系统的非功能测试,并且从系统运行效果的展示中可以看出,该系统交互体验良好、功能有效,已经满足基本业务需求与系统功能需求,总体来说符合课题预期目标。

第六章 总结与展望

6.1 课题完成情况总结

随着互联网行业的高速发展与规模的日趋庞大，网络完全融入了人们生活的各个方面，人们对信息获取需求暴增，产业转型数字化的进程也在不断推进，在后疫情时代，人们在医疗健康方面的需求日益增加。而在用户提出了医疗类问题之后，国内几种常用的搜索引擎通常只会将与问答文本中有一定相关性的网页链接给予用户展示，导致无法直接有效地给出精确的回答。基于国家智慧医疗项目数字化推进的背景，本文最终完成了针对医疗垂类下的智能问答系统的设计与实现。当前课题完成的工作情况如下：

1) 完成了基于 BERT 和 BM25 的智能问答系统的设计

通过对智能问答系统和智能问答算法的相关资料文献的查询与研究，拆解出当前在课题任务执行的过程中，存在的问题与挑战，并针对其中具体的问题点进行详细的可行性分析与需求分析，最后基于需求分析提出了一套完整的基于 BERT 和 BM25 的智能问答系统的概要设计与详细设计方案。

2) 完成了基于 BERT 和 BM25 的智能问答系统实现

本课题基于系统的概要设计与详细设计，对智能问答系统完成了具体实现。其中在核心算法部分，以“召回-排序”为整体流程，基于 BM25 算法，实现了一种字面解析的召回方法；同时本文基于 BERT 预训练模型，实现了一种语义解析的排序方法。具体来讲，算法会先根据用户问句的字面分析方法，进行初步相似文本筛选，再通过语义分析方法进行最优排序，得到与用户问句最相似似的候选问句，最终通过该候选问句查找到对应的答案。

3) 完成了基于 BERT 和 BM25 的智能问答系统测试

根据基于 BERT 和 BM25 的智能问答系统的设计流程，结合具体的基于 BERT 和 BM25 的智能问答系统的实现工作，完成了整个系统的测试——经测试，该系统交互体验良好、功能有效，能够达到人们对日常医疗保健相关问题的查询需求满足的标准，总体来说符合课题预期目标。

6.2 课题未来工作展望

本课题研究在基于 BERT 和 BM25 的智能问答系统研究与实现中，仍然存在

着一些问题待进一步解决或优化，具体的改进工作体现在如下几个方面：

其一，本文提出的基于 BERT 和 BM25 的智能问答系统可以从功能上与性能上再进行进一步的优化，具体优化方向为：在功能上，主要是针对于智能问答系统的排序层，当前使用的是 BM25 向量来得到文本的字面解析结果表征，一定程度上存在表达能力不足的问题，下一步的研究思路是训练基于双塔结构的神经网络模拟，来达到更深度表征文本向量的效果，同时配合使用最近邻居查找算法提升召回性能与效果；在性能上，主要是针对于智能问答系统的排序层，当前使用的是微调的 BERT 预训练模型来得到文本的语义解析结果表征，其参数量与运算推理耗时较大，本质上是由于模型参数过多，导致计算量巨大，下一步的思路是通过知识蒸馏的方法对 BERT 预训练模型进行参数缩减，在不影响模型整体预测效果的情况下，提升系统模型的运算效率。

其二，本文实现的基于 BERT 和 BM25 的智能问答系统，目前仅在医疗健康领域有应用，本文提出的系统设计方案在行业领域间有高度的可移植性，因此未来可以根据不同领域的高质量语料、新兴数据集等进行领域方面的系统迁移，在不同行业方向诸如通信领域的智能聊天问答、电商领域的智能客服问答、科研领域的智能问答等，充分发挥其创新作用。

其三，本文提出的基于 BERT 和 BM25 的智能问答系统功能比较单一，未来可以加入如医院挂号系统、医疗科普知识推荐系统、购药商城系统等子系统或者功能，结合本文提出的系统组成一个完整的应用软件，提升用户的使用体验与软件产品的效用。

6.3 软件工程职业素养认识

在课题任务实施研究的过程中，个人认为对软件工程职业素养进行一定的学习培养也是不可或缺的，这样才能保证在个人职业发展中始终保持行业领军人才的专业性。作为一名软件工程师，本人在软件工程职业素养有着一定程度的认识，总结如下：

1) 其一，在职业道德与规范方面：

首先，不可故意破坏软件功能，失误率保持在一定水平之下是一位专业的程序员的责任，而且不要轻易使用明知有缺陷的代码，要确保相应测试通过之后再上线才是合格的软件工程师的风格；其次，要坚持代码设计原则，在代码编写过程中要保持良好的软件代码结构以及编码风格；再者，对内对外要保持一名专业的软件工程师的自信，但不要过度自信导致自负，自负越大盲点越大；同时，信

守承诺是一名严谨负责的专业软件开发人员应该需要具备的素质；最后，作为一名软件工程师，需要始终维护国家法律法规，不越过红线滥用用户信息，必须保证所开发的软件不会对社会造成不良影响甚至危害。

2) 其二，在软件编程与职业健康方面：

首先，需要保证自己编写代码能够在系统环境下正常运行，同时具备较高的可读性与对应的文字注释；其次，始终贯彻以健康为本的思想，需要平衡自己的生活方式与工作节奏，保证自己的身体健康，这样才能在工作中尽心尽力，在生活中保持活力，编写出高质量的程序；同时，作为专业的软件工程师，在工作之余也需要在自己所在的领域进行广泛的阅读，只有通过不断的学习，才能保持自己不断的进步与创造力，坚持自己在工作中乃至行业中承担贡献者的一个角色；最后，对于实际问题需要发挥软件工程的思想，从多方面对问题进行分析，不要一直研究问题本身，要剖析问题本质，这样有助于自我职业发展与职业健康。

致谢

时光如梭，转眼间本科四年的光阴已经过去，一路以来虽是磕磕绊绊，但在过程中收获颇多，也算有所成长，不忘初心，不负青春。

首先，要感谢我的导师吴劲副教授。本论文的工作是在吴劲老师的悉心指导下完成的，从本科毕业设计定题之初，吴劲老师便事无巨细地指导我课题研究的方向与课题目标，并时常督促我完成论文的撰写，在初稿修改期间，吴劲老师也给出了宝贵的修改意见，使得我的论文撰写比较顺利，学术写作水平有很大提高，在此向老师致以我崇高的敬意与真挚的感谢。

同时，要感谢电子科技大学信软学院的教职工工作者，感谢你们为我们学院同学打造了一个良好的学术氛围以及实践氛围，特别是在实习的时间里，我在不断的实践过程中收获颇丰。

另外，感谢辅导员以及我的室友和学院微光工作室同学们，感谢你们在生活上与学习上提供的帮助，让我的本科四年时光多姿多彩，留下美好的回忆。

最后，感谢我辛勤的父母，感谢你们对我的关心与支持，以及所做的无私奉献。

感谢所有帮助过我的人，在这里真诚的对你们说一声谢谢！

参考文献

- [1] Toyhom.Chinese medical dialogue data 中文医疗对话数据集 [CP].<https://github.com/Toyhom/Chinese-medical-dialogue-data>.Dec 24, 2019
- [2] Jacob Devlin, Ming-Wei Chang, Kenton Lee and Kristina Toutanova.BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding[C].North American Chapter of the Association for Computational Linguistics, 2018, 1-7
- [3] Stephen Robertson and Hugo Zaragoza.The Probabilistic Relevance Framework: BM25 and Beyond[M].2009, 347-370.
- [4] Chujie Zheng, Yunbo Cao, Daxin Jiang and Minlie Huang. Difference-aware Knowledge Selection for Knowledge-grounded Conversation Generation[J].arXiv: Computation and Language, 2020: 3-8.
- [5] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi and Hannaneh Hajishirzi. Bidirectional Attention Flow for Machine Comprehension[C].international conference on learning representations, 2016, 2-6.
- [6] Tomas Mikolov, Kai Chen, Greg S. Corrado and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space[C].international conference on learning representations, 2013, 3-5.
- [7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser and Illia Polosukhin.Attention is All you Need[C].neural information processing systems, 2017, 2-6.
- [8] Sun Junyi.“ ‘ 结巴 ’ 中文分词：做最好的 Python 中文分词组件 ” [CP].<https://github.com/fxsjy/jieba>.
- [9] Jina AI.CLIP-as-service[CP].<https://github.com/jina-ai/clip-as-service>.May 4, 2022

外文资料原文

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Jacob Devlin Ming-Wei Chang Kenton Lee Kristina Toutanova

Google AI Language

{jacobdevlin, mingweichang, kentonl, kristout}@google.com

Abstract

We introduce a new language representation model called **BERT**, which stands for **Bidirectional Encoder Representations from Transformers**. Unlike recent language representation models (Peters et al., 2018a; Radford et al., 2018), BERT is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task-specific architecture modifications.

BERT is conceptually simple and empirically powerful. It obtains new state-of-the-art results on eleven natural language processing tasks, including pushing the GLUE score to 80.5% (7.7% point absolute improvement), MultiNLI accuracy to 86.7% (4.6% absolute improvement), SQuAD v1.1 question answering Test F1 to 93.2 (1.5 point absolute improvement) and SQuAD v2.0 Test F1 to 83.1 (5.1 point absolute improvement).

1 Introduction

Language model pre-training has been shown to be effective for improving many natural language processing tasks (Dai and Le, 2015; Peters et al., 2018a; Radford et al., 2018; Howard and Ruder, 2018). These include sentence-level tasks such as natural language inference (Bowman et al., 2015; Williams et al., 2018) and paraphrasing (Dolan and Brockett, 2005), which aim to predict the relationships between sentences by analyzing them holistically, as well as token-level tasks such as named entity recognition and question answering, where models are required to produce fine-grained output at the token level (Tjong Kim Sang and De Meulder, 2003; Rajpurkar et al., 2016).

There are two existing strategies for applying pre-trained language representations to downstream tasks: *feature-based* and *fine-tuning*. The feature-based approach, such as ELMo (Peters et al., 2018a), uses task-specific architectures that include the pre-trained representations as additional features. The fine-tuning approach, such as the Generative Pre-trained Transformer (OpenAI GPT) (Radford et al., 2018), introduces minimal task-specific parameters, and is trained on the downstream tasks by simply fine-tuning *all* pre-trained parameters. The two approaches share the same objective function during pre-training, where they use unidirectional language models to learn general language representations.

We argue that current techniques restrict the power of the pre-trained representations, especially for the fine-tuning approaches. The major limitation is that standard language models are unidirectional, and this limits the choice of architectures that can be used during pre-training. For example, in OpenAI GPT, the authors use a left-to-right architecture, where every token can only attend to previous tokens in the self-attention layers of the Transformer (Vaswani et al., 2017). Such restrictions are sub-optimal for sentence-level tasks, and could be very harmful when applying fine-tuning based approaches to token-level tasks such as question answering, where it is crucial to incorporate context from both directions.

In this paper, we improve the fine-tuning based approaches by proposing BERT: **Bidirectional Encoder Representations from Transformers**. BERT alleviates the previously mentioned unidirectionality constraint by using a “masked language model” (MLM) pre-training objective, inspired by the Cloze task (Taylor, 1953). The masked language model randomly masks some of the tokens from the input, and the objective is to predict the original vocabulary id of the masked

外文资料译文

BERT：深度双向变压器的预训练语言理解

三. BERT

我们在本节介绍 BERT 及其详细实现。我们的框架有两个步骤：预训练和微调。在预训练期间，该模型通过不同的预训练任务在未标记数据上进行训练。对于微调，BERT 模型首先使用预训练的参数进行初始化，然后使用来自下游任务的标记数据对所有参数进行微调。每个下游任务都有单独的微调模型，即使它们使用相同的预训练参数进行初始化。图 1 中的问答示例将作为此示例的运行示例。

BERT 的一个显着特点是其跨不同任务的统一架构。预训练的架构和最终的下游架构之间的差异很小。

模型架构 BERT 的模型架构是基于 Vaswani 等人描述的多层双向 Transformer 编码器(2017)并在 tensor2tensor 库中发布。由于 Transformers 的使用已经变得普遍，而且我们的实现几乎与原来的相同，我们将省略对模型架构的详尽背景描述，并请读者参考 Vaswani 等人（2017 年）以及“带注释的变压器”等优秀指南。

在这项工作中，我们用 L 表示层数（即 Transformer 块），用 H 表示隐藏大小，用 $A.3$ 表示自注意力头的数量我们主要报告两种模型大小的结果：

BERTBASE($L=12, H=768, A=12$, 总参数=110M)和 BERTLARGE($L=24, H=1024, A=16$, 总参数=340M)

BERTBASE 被选为具有与 OpenAI GPT 相同的模型大小以进行比较。然而，至关重要的是，BERTTransformer 使用双向自注意力，而 GPT Transformer 使用受约束的自注意力，其中每个令牌只能参与竞争

.....