

基于 Web 的英文智能问答系统实现策略研究

陈 钰 王迎春 牛苗苗

(唐山科技职业技术学院, 河北 唐山 063000)

摘 要 随着互联网的普及, 智能型设备, 如手机、平板电脑、笔记本电脑的应用随处可见, 英语学习者们越来越热衷于在网上进行自主学习。本文通过设计英文智能问答系统, 充分整合网络资源, 以对话的交互界面为窗口向提问者提供拟人化的智能回答。本系统主要基于英语语言处理的相关技术以及海量的英语交流数据库, 它可以在对提问者进行自动化回答后, 分析提问者反馈信息, 对于不满意的答案, 系统会转给相应的专业人士进行处理。在这个交互过程中还储存了相应的知识回答轨迹, 并指向相应的知识库链接, 方便后来的用户进行查看。

关键词 Web 英文智能问答系统 交互界面 测试环境

中图分类号: TP311.52

文献标识码: A

文章编号: 1007-0745(2022)06-0026-05

1 功能需求

基于 Web 的英文智能问答系统运行的首要步骤是先获取到提问者的问题。之后, 根据匹配模型将该问题与数据库中所有的相关问题进行匹配, 以找到最优的答案并将其反馈给提问者。而对于出现在知识图谱的相关语法问题, 问答系统则还需要识别句子中出现的语法结构, 并调用不同的场景模式进行响应预测与问答交互^[1], 因此该智能问答系统需要具备的功能有:

1. 英文问答数据获取: 利用 Python 等技术从各个公开可获取的如 Google、Wiley InterScience、Ingenta 等网站、论坛等自动获取相应的问答数据。

2. 英文相似问题制备: 获取到的中文问答数据对, 以相似性原则制备问题数据集, 用于文本进行相应的匹配模型转译。

3. 相似问题更迭匹配模型: 对于新获取的问答数据, 生成新的相类似问题和问答系统自身积累的对话模型数据对相似问题进行更迭匹配。

4. 知识图谱构建: 基于现有的英文数据库自动构建相应的知识图谱。

5. 前台对话式交互: 可以模拟提问者和回答者的交互, 通常以前台交互界面, 以对话的方式进行呈现, 可以通过类似微信一问一答的交流方式, 进行问题的解决。并在最后获得用户的评价, 可以以此来判定返回的模型的正确性以及相似的问题, 是否被提问者所接受, 进而来决定是否用上条功能中的模型进行更迭匹配。

6. 模型的调用: 可以根据用户的需求对相应的模型予以响应。

7. 实体知识库链接: 对于在问答中出现的较为重要的实体内容, 可以与实体知识库进行链接, 方便用户对于实体进行更直观、更便捷的访问和查看。

2 英文智能问答系统的架构设计

英文智能问答系统的总体框架包括代码层、数据库层、应用层以及用户层(具体如图 1 所示)^[2]。代码层表示智能问答系统所涉及的所有编程语言与软件。数据层通过人工、Python 等数字技术获取来自各大英文网站、已发表文献的公开文献数据以及英文问答数据, 其次也从各英文网站采集英文实体知识库, 例如名词解释等, 均是从比较权威的数据库中进行采集。

英语同义问题数据集是通过数据采集完成后使用同义语料制备平台依据数据库中的所有问答题目形成的。应用层是由数据库层所收集的丰富问答数据库和详实的知识库及其产生的同义问题集合体共同形成的具备智能问答模式的应用服务类模型。自动问答服务可以根据类似问题归集的衍化形成集演练、推测、数据更迭匹配等相类似的模型匹配。而且还能够根据实体认知图谱进行反向预测模型推演, 改进和提升问答效率^[3]。在推演过程中形成的副产品词汇向量和文本向量可以模型化后运用于其他的任务, 形成服务类模型的基础。应用层的文本分词功能可以形成导向索引, 为后期提供实体链接指向。不同的接入口面对的是庞大的用户层, 可以根据用户通过提出的问题, 实现相类似问题解答、疑难问题专人解答以及实体知识指向链接等功能。通过数据服务可以根据提问者对回答结果的满意度来进行评判, 以确定是否需要通过专人解答或者模型的更迭功能对非同类问题予以处理。在用

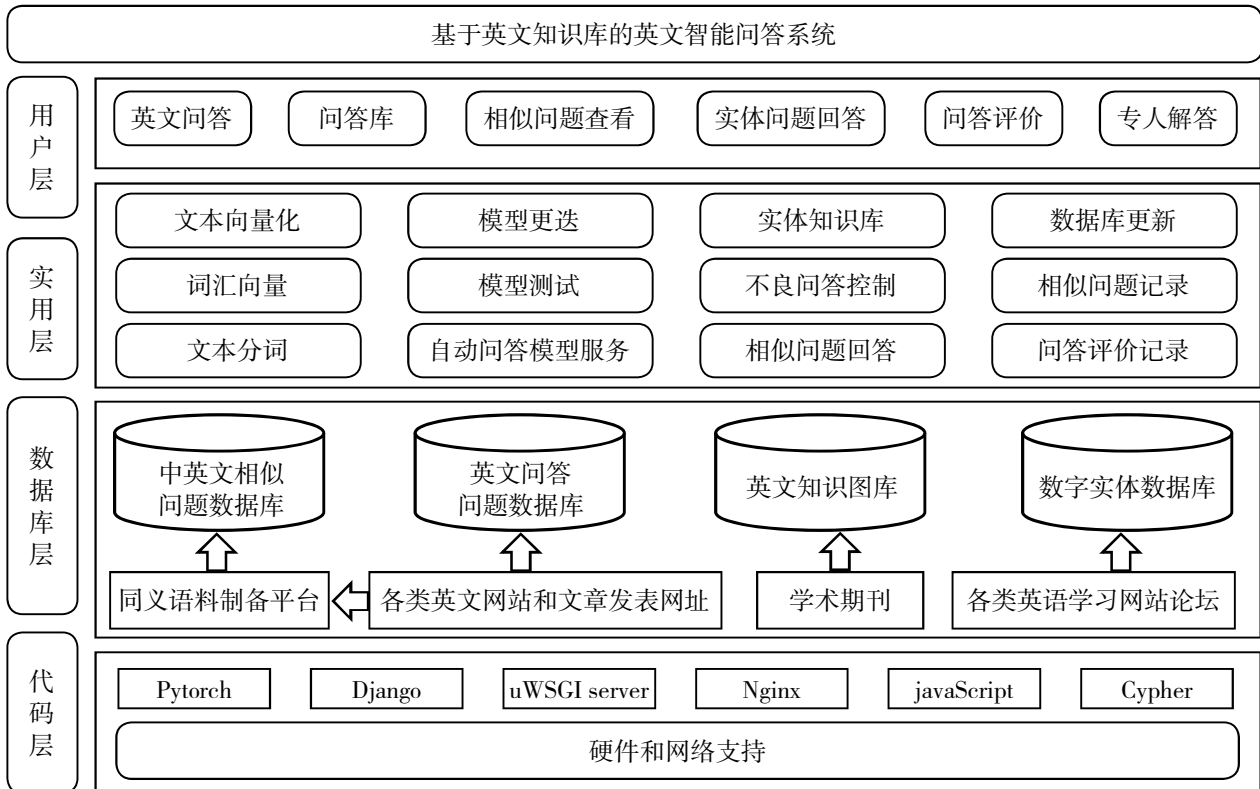


图1 基于英文相似问题匹配的智能问答系统架构图

户层,可以实现在提问者之间的交互,获得相应的问题,经过应用层数据库的调用,搜索相应的答案。在用户提出有代表性的问题且能够指向实体知识库链接的同时,提问用户可以根据系统所提供的实体库信息链接进行点选以获取更多的有关信息。

3 系统核心技术

智能问答系统的代码层所采用的代码包括Python、Django、PyQuery、Nginx、uWSGI、Matchzoo等,后台模型主要通过Python实现,Nginx、uWSGI和Django组合方案用于线上项目的部署。Matchzoo属于Python工具包,用于各种文本模型匹配,且可以把这些模型用统一的接口封装。其优点在于便于文章的检索,以及对于问题进行自动回应和答复,对于同义语句,可以达到快速拣选和识别,从而能够较为快速和精准地进行个人数据的调试,在很大程度上节省了开发时间。自定义模式下,文本可以用相似问题匹配的方式在该架构内实现。PyTorch作为具有开放意义的深层次研学架构,由于其设计与人类思维力相匹配,在语句处理方面的表现令人钦佩,它可以让用户在形成自身想法的同时保持更加专注的状态,能够更加适合深度和高效的学习。它与Matchzoo工具包相结合调用能够实现

类似于人类神经网络的自主智能问答反馈。uWSGI是用于和Web应用在网络架构中数据交换的服务器。Nginx是较uWSGI简单的一款服务器,职能是负责统管理所接收到的所有浏览器需求。它可以自行处置由浏览器发送的静态请求,而动态请求则由其分发到uWSGI在Web应用上处理并做出响应。Django可以保证后台Web应用在规范的框架下运行。PyQuery作为Python的一项巧妙灵活的网页解析工具,被大多数开发者所青睐,目的是能够在指定的网页节点中得到所需要的数据。

数据库层负责收集、筛分、存储和重组问答数据、查找实体知识库和英文知识图谱和类似问题数据的比对与更新。在解决用户的提问时,系统首先经过文献查找,搜寻曾经被公开的问答数据库相应的字符,并将其以文本的形式快速导入数据库。之后,从各个英文类网址上进行相应的采集,将所输入的文本字段加以横向比对,以确定与本次查找相匹配的字段并进行收集,然后用Python技术黏贴网页代码并利用PyQuery解析并保存。系统在访问网站之后,通过获取新的问答数据,将已有的数据库进行更新。对相关的问题数据进行分离,利用语料制备模型生成相对应的数据集,正样本和负样本按照一定的比例生成,得到相似

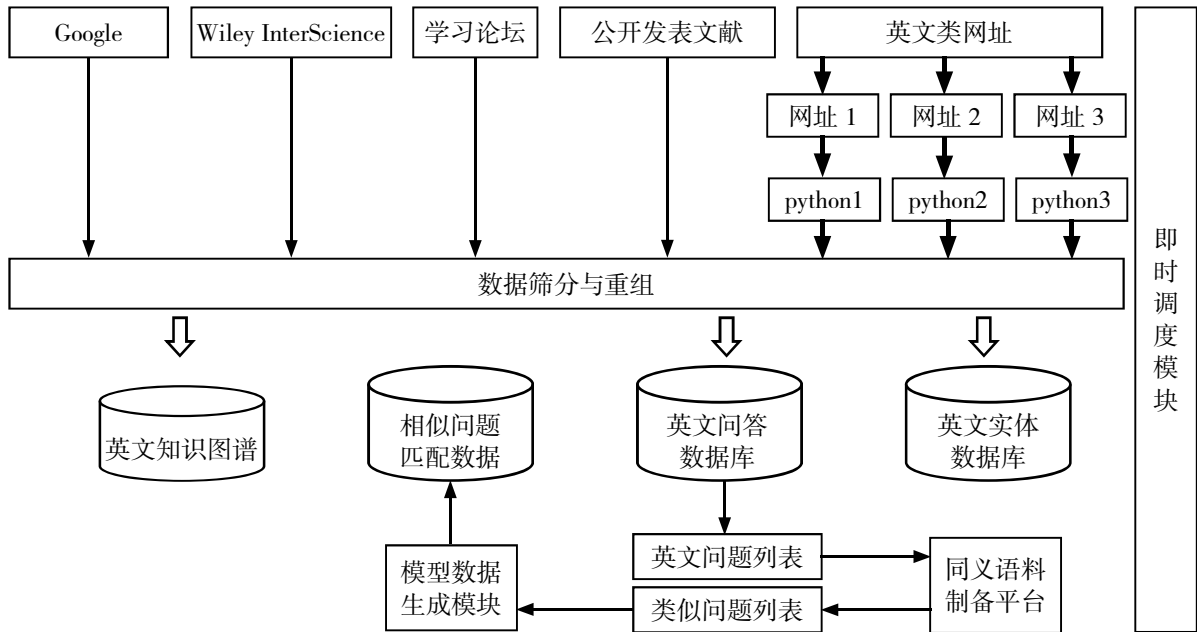


图2 英文智能问答系统数据获取与处理流程

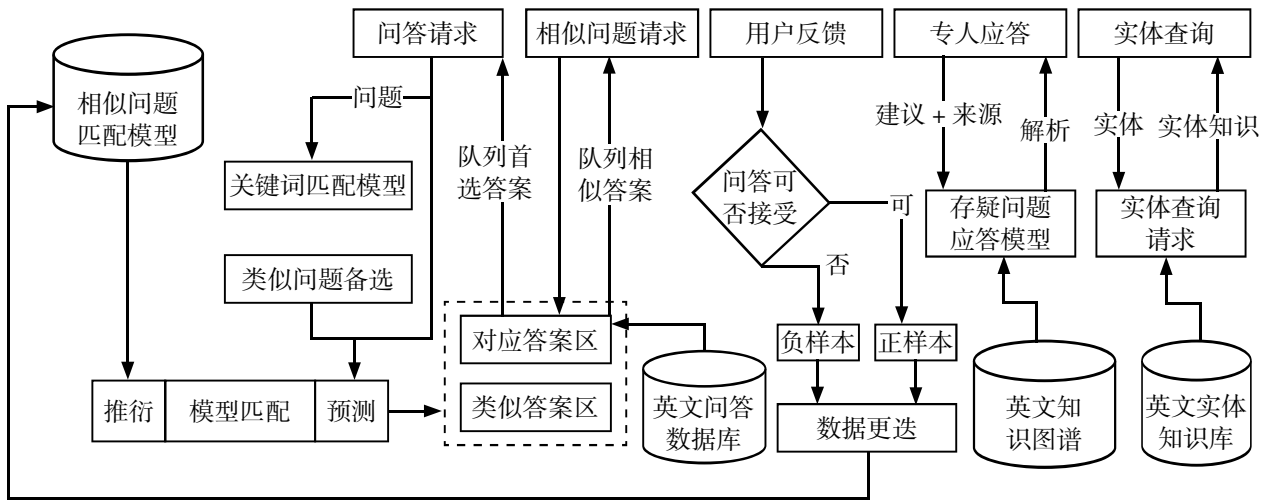


图3 英文智能问答系统应用层各API响应流程

的问题答案后用于填充匹配语料模型。对于英文知识库可以直接从 Google 和 Wiley Inter Science 获得，通过筛分和实物比对后可以生成英文知识库图谱。英文智能问答系统数据获取与处理流程如图 2 所示。

应用层模型建立于 Python 语言之上，用 Pytorch 搭建框架。主要用于数据库数据生成后的问答与反馈，它可以直接调用集成在 Django 应用中的数据模型。应用层在调用由 API 传达的用户提问时，会率先采用匹配关键词和筛选模型被选问题集合的方式。之后用事先排布好的深层次匹配模型查找相对应的与选择项最

相近的问题，并迅速反馈给用户层。用户层可以再一次启用类似问题的 API 问询，应用层可以依照之前的排布，选取最优的回答选项，按照不同的设置配比，返还给用户层。如果用户层对于返还的回答选项不满意，则可以启用存疑问题应答模型，让用户来决定是否采用专人应答模式。如果用户单纯采用实体查询模型，即可以在英文实体知识库中搜索相应的实体答案，返还给与其对应的文本模块。应用层与用户层的每一次交互都附带有来自用户层的反馈。如查询用户得到了可接受的答案，即为给出了提问者相对应或者相类

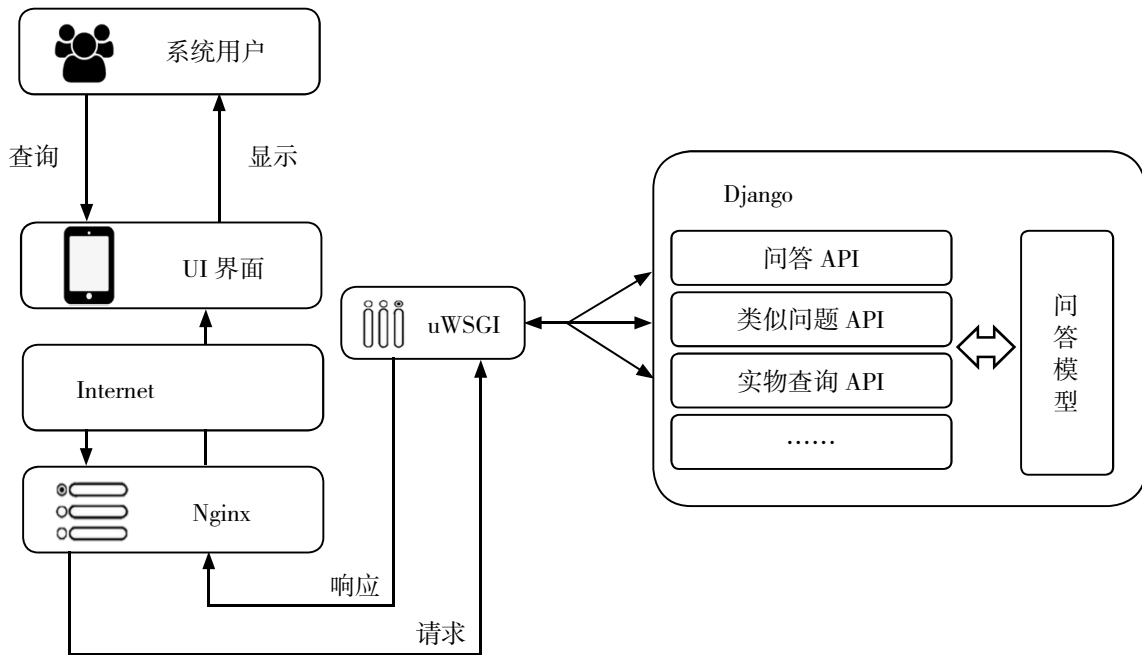


图4 英文智能问答系统线上部署架构示意图

类似的信息。应用层会将这些信息进行分组，及时的记录到已有知识数据库中作为正样本储存。如果没有得到用户层的正向反馈，记录将称为负样本，采用数据更迭的方式来寻求相似问题的匹配模型进行下一个循环。英文智能问答系统应用层各API响应流程如图3所示。用户层与系统的前台进行直观的交互，对应层的API封装后，准备前段进行访问。Nginx可以直接处理前台的静态问题，可以用链接的形式将问题指引到相应的静态文件模型，并将结果发送到uWSGI，经过调用Django将处理过的信息回应给反馈系统。前台UI界面可以简洁明快地表达交互界面，Javascript以及jQuery在开发中将要被用到。

拟建立的英文智能问答系统的开发部署现处于内网测试阶段，待后续成熟可以直接平移到外网服务器。前台的架构部署如图4所示，访问者可以通过设备的浏览器直接进入问答交互界面，在提出相应的问题后，由系统前端发送请求，Nginx根据接收到的信息调用uWSGI中的Django对发起的请求做出回应。类似问题在查询请求方面和实体查询的过程基本一致，前台在接收到后台的响应之后，通过直接显示或是页面跳转，实现前后台之间的交互。

4 英文智能问答系统界面展示

4.1 交互界面的样式

使用者可以通过设备的浏览器或是微信小程序进入英文智能问答系统界面，交互方式用类似于微信聊

天的形式进行。使用者可以输入一定量问题并点击“Send”按钮发送，前台在获取信息转入到后台调用回复资源后，将以对话的形式将答复传递给使用者。

以问题“Interview”为例，使用者在输入并点击发送“Send”按钮后，系统回复为：

- “1.Purpose of Interviews
- 2.Types of Interviews
- 3.Preparing for the Interview
- 4.Interview Structure.
- 5.Typical Questions”

由于该问句所提供的信息不够清晰。所以，系统以选项的方式予以呈现，用户可以继续在选项上点选，跳到下一详细页面。当点选“4.Interview Structure.”，页面内容跳转，系统的回复为“Traditional:Traditional interview questions focus on your educational background, work experience, activities,skills,etc.Behavioral:Behavioral interviews focus on specific past behaviors”，Describe a time when.Past behavior is the best predictor of future performance. 在每轮对话完成后，使用者可以根据对话框下面的评价按钮对本次回答情况进行点选评价。当使用者感到满意时，可以点选相应的按钮，平台会自动将被点选为满意的答案归置为类似问题，并将其用正样本的方式添加到与英文知识库相匹配的模型中。若使用者对此项回答点选不满意，该项记录则被列为负样本。当数据量累积到所设置的限度时，系统将自

表1 英文智能问答系统测试内容与结果

测试步骤	是否通过	平均响应时间
(1) 移动设备端进入首页。	是	0.5 秒
(2) 发起问题, 查看系统是否能够及时回复。	是	0.5 秒
(3) 点击相似问题, 查看系统能否跳转下个相似问题。	是	0.1 秒
(4) 点击实体链接, 查看系统是否跳转实体对应解释。	是	0.1 秒
(5) 对当前回答点击“Satisfied”, 查看后台是否记录相似问题。	是	0.5 秒
(6) 对当前回答点击“Dissatisfied”, 查看后台是否记录不相似问题。	是	0.5 秒
(7) 对当前回答点击“More”, 查看后台是否记录不相似问题, 能否提供更多相关信息并做出响应。	是	0.5 秒

动进行模型更迭, 以改进和提升模型的整体效果。

在系统回答提问者的问题之后, 使用者还可能对某些具体的细节有所疑问, 所以系统配备了链接词汇查询功能。系统所给出的所有答复都是在后台实体库中备案并添加了链接的, 所以使用者可以在对话框中把不同颜色进行区分的可点选内容做出进一步的查看。

4.2 类似问题点选界面

使用者对系统所作出的回答不满意, 还可以点击会话框下的“More”按钮, 系统即可跳转到下一个有可能与提问相类似的问题, 并作出相应的回答。如再次给出的答案, 不能满足使用者的需求, 使用者则可以点选专人回答的方式, 届时将有相应领域的专家通过远程问题解答方式予以解决, 由于该系统目前正处于测试阶段, 所以系统会回复“Thanks for your question”。使用者可以通过搜索系统浏览器的方式寻求其他解决方案。

5 系统性能测试

5.1 测试的意义

系统在上线之前要经过不同层次的联合测试, 测试主要有两个方面:

一方面是功能测试, 主要是为了使开发者了解该智能问答系统是否满足最初的设计需求。测试的内容包含了问题回答的完整性、相似问题的查看是否简单、快速、便捷, 系统运行中链接指向是否有难以跳转、卡顿或是出现页面乱码的现象、显示字体的大小、颜色的深浅度、接受度等等。

另一方面是通过特定的工具, 对于真实使用环境下的系统反应速度予以精确的测定。

5.2 测试环境

测试环境分为开发者前段服务器端和测试应用条件下的电子设备移动端。

5.3 测试内容与结果

测试内容: 系统启动是否正常、问答功能是否齐全、相似问题是否可查询、实体库知识查询、会话结果评价反馈是否正常等。测试步骤和结果见表1。

测试结果: 对系统进行功能测试, 各项功能均能正常实现。在需要进行问题回答的步骤时, 由于需要调用关键词汇进行模型筛选验算, 所以耗时比其他的功能稍长, 但是也在可以接受的范围。

6 结语

本论文主要阐述了基于智能问答模型的英文智能问答系统的设计与实现。虽然该系统仍处在不断调试完善的阶段, 但是笔者希望通过对涉及的系统搭建方面的关键技术的掌握和实施, 今后能在市面上看到更多不同种类的智能问答系统设计程序, 以便更好地用科技的力量服务于社会。

参考文献:

- [1] 谢项. 基于古诗知识图谱的智能问答研究[D]. 武汉: 华中师范大学, 2020.
- [2] 王松嘉. 基于机器人知识图谱的智能问答系统的设计与实现[D]. 哈尔滨: 哈尔滨师范大学, 2020.
- [3] 王璐. 基于知识图谱的健康膳食知识智能问答系统[D]. 兰州: 兰州大学, 2020.