湖州师范学院

2023 届毕业设计(论文)

课 题 名 称: 面向职业生涯规划的智能问答系统研究

课 题 名 称(英文): Research on intelligent question answering system for career planning

学 生 姓 名: 廖 义 学 号: 2019162403

专 业 名 称: 计 算 机 科 学 与 技 术

指 导 教 师: 吴 茂 念 职 称: 教 授

所 在 学 院: 信 息 工 程 学 院

完 成 日 期: 2023 年 4 月 27 日

教务处制表

面向职业生涯规划的智能问答系统研究

摘要：随着社会和经济的发展，职业规划已成为越来越多人关注的话题。然而，对于大多数人来说，职业规划仍然是一个难以解决的问题。为了解决这一问题，本文研究了一个面向职业生涯规划的智能问答系统，并探讨了相关技术。该系统采用基于知识图谱的问答技术，能够自然语言理解和语义分析用户提出的职业信息问题，并从知识图谱中获取相关知识，提供准确全面的答案。此外，该系统还包含职业测评，以初步推荐适合用户的职业选择，为用户提供个性化的职业推荐。

本文研究的系统主要技术包括自然语言处理、知识图谱和推荐算法。自然语言处理技术基于 jieba 库和 gensim 库进行词性标记和停用词过滤，用于语义分析和理解用户提出的问题。知识图谱的构建基于 Neo4j 库和 py2neo 库，用于存储和管理职业规划相关的知识和信息。推荐模块基于 Scikit-learn 库和 TensorFlow 库，用于根据用户的自身情况进行个性化推荐。未来的研究可以结合机器学习，用于刻画用户画像和对用户的历史问题和答案进行分析，以进一步提高系统的推荐精度。

关键词：智能问答系统；职业规划；知识图谱；机器学习；职业测试

Research on Intelligent Question Answering System for Career Planning

Abstract:

With social and economic development, career planning has become an increasingly popular topic of interest. However, for most people, career planning is still a difficult problem to solve. To solve this problem, this paper investigates an intelligent question and answer system for career planning and discusses related technologies. The system uses knowledge graph-based question and answer technology, which is capable of natural language understanding and semantic analysis of career information questions posed by users, and provides accurate and comprehensive answers by obtaining relevant knowledge from the knowledge graph. In addition, the system includes a career assessment to initially recommend suitable career options for the user and provide personalised career recommendations for the user.

The main technologies of the system studied in this paper include natural language processing, knowledge graphs and recommendation algorithms. The natural language processing technology is based on the jieba and gensim libraries for lexical tagging and deactivation filtering, which are used for semantic analysis and understanding of the questions posed by the user. The knowledge graph is built on the Neo4j and py2neo libraries for storing and managing knowledge and information related to career planning. The recommendation module is based on the Scikit-learn library and the TensorFlow library and is used to personalise recommendations based on the user's own profile. Future research could incorporate machine learning for user profiling and analysis of users' historical questions and answers to further improve the system's recommendation accuracy.

Keywords: Intelligent Q&A system; career planning; Knowledge graph; machine learning； Career test

目 录

第一章 绪 论

随着人工智能技术的持续进步，以自然语言形式直接完成与数据库或知识库的问答式交互将代替关键词搜索成为新的发展趋势、人机交互的主要形式之一以及人类获取数据与知识的主要入口形式之一。本章节主要介绍面向职业规划的智能问答系统的研究背景及意义、研究现状和发展趋势及本文主要研究内容。

1.1 研究背景及意义

日前“双减”政策的推行，应试教育逐渐退出了舞台，素质教育提上了日程。与应试教育相比，素质教育更加重视人的思想道德素质、心理素质、能力培养等。时代不断进步，我国经济结构日趋完善，社会分工愈来愈精细标准化，社会对于各方面优秀人才的需求也日渐增加。随着社会的发展和经济的变化，职业规划已经成为越来越多人关注的话题。职业规划不仅关系到个人的发展和成长，也关系到社会的稳定和发展。然而，由于职业规划涉及到众多领域的知识和信息，国家且传统教育中“生涯教育”部分有所欠缺，对于大多数人来说，职业规划仍然是一个难以解决的问题。[1]

随着互联网的快速发展，信息越来越多、越来越繁杂。用户在进行搜索时，可能会因为搜索结果过多、过于相似等原因导致大量的时间、精力被耗费，仍然难以找到所需的信息。传统搜索算法通常基于关键词嵌入式的信息检索方式，这种方式仅仅关注关键词出现的次数、位置等因素，而未能充分考虑词义、语义、语境等多方面的信息，因此难以准确把握用户的检索意图。另一方面，随着信息技术和制造技术的飞速发展，大量的虚假和相似信息也随之出现，这就更加增加了传统搜索算法的困难度。解决这些问题，需要发展新的搜索技术，例如基于自然语言处理、机器学习等技术的智能问答系统，能够更加精准地识别用户的检索意图，分析搜索结果的多个维度，提供更加丰富、准确的信息。

面向职业规划的智能问答系统可以帮助用户了解自己的职业发展方向，提供相关的职业规划建议和资源。本课题主要研究面向职业生涯规划的智能问答系统，通过分析各类智能问答系统技术的应用，研究如何实现一个初级的职业规划的智能问答系统，根据用户的问题，以及用户的相关信息或测评结果提供相关的职业规划建议或职业信息，帮助用户了解自己的职业发展方向。

且传统的职业规划服务相比，面向职业规划的智能问答系统有许多优点。首先，它是全天候的。智能问答系统可以随时随地进行在线查询和回复，也就是说，即使在非工作时间，用户也可以使用这个系统寻求帮助。其次，它是高度个性化的。智能问答系统可以针对每个用户的需求提供特定的答案，因此提供的信息往往更具有价值。更重要的是，它是高效的。在传统的职业规划服务中，用户可能需要花费较长时间才能获得有用的答案。而智能问答系统在短期内就能够快速回复用户的问题，使用户能获得较为精准的答案。

1.2 研究现状及发展趋势

本节主要介绍职业规划在国内外的研究、智能问答系统研究现状、面向职业规划的智能问答系统研究现状和发展趋势。

1.2.1 职业规划研究现状

近些年，国家陆续制定了有关职业教育发展的战略，其中以文件《职业教育提质培优行动计划(2020—2023 年)》的出台作为标志性事件，这些政策的颁布为职业教育的成功转型提供了契机，为职业教育的快速发展打下了基础，保障了我国对于高素质、高技能职业人才的培养。在此基础上，越来越多的广大群众就已经开始专注到职业教育，国家也开始鼓励大家积极投身于发展社会主义经济。

现有相关生涯规划研究主要聚焦于大学毕业生的职业生涯规划及就业指导。国内学业生涯规划教育主要局限于职业技术院校和本科院校, 在中学阶段还鲜有涉及。而国外在这方面的研究及执行模式上，学业生涯规划地执行形式是以合作模式为基础的，专业的学业规划师为了全面了解学生，帮助其发展，需要与教学人员、家庭、机构、企业等人员建立联系、保持沟通。

1.2.2 智能问答系统研究现状

自动问答系统的技术发展历程可以追溯到上世纪 50 年代，当时图灵测试被提出作为判断一个计算机是否具备人类智能的标准。在此之后，研究者们开始探索如何通过自然语言处理、信息检索、机器学习等技术来构建更完善的自动问答系统。在 80 年代和 90 年代，自动问答系统开始应用于一些特定的领域，如医学、法律、金融等。这些系统通过专业领域的语言模型和知识库，能够提供更为精准的问题解答。而直到 21 世纪初，自动问答系统才逐渐开始应用于更广泛的领域，如搜索引擎、智能语音助手、智能家居等，使用户的信息交互更方便快捷。

智能问答系统基于人工智能技术，它可以通过自然语言处理、知识图谱、机器学习等技术，实现对用户提出的问题进行智能化回答。目前，智能问答系统已经在多个领域得到了广泛的应用，如智能客服、智能教育、智能医疗等。

智能问答系统是人工智能领域的一个重要研究方向，目前已经有很多研究者在这个领域做出了很多有意义的工作。其中，基于知识图谱的智能问答系统是当前研究的热点之一，其主要思想是将知识图谱中的实体和关系作为问题和答案的依据，通过自然语言处理技术将问题转化为查询语句，再利用知识图谱中的信息进行答案的推理和生成。此外，基于深度学习的智能问答系统也是当前研究的重点之一，其主要思想是通过深度学习技术学习问题和答案之间的映射关系，从而实现智能问答的功能。近年来在国内外都得到了广泛的关注和研究。国内的研究主要集中在基于知识图谱和深度学习的问答系统，如百度的 Duer、阿里的阿里小蜜等。而国外的研究则更加注重多语言、跨领域和多模态的问答系统，如 IBM 的 Watson、微软的 Satori 等。此外，还有一些研究关注于问答系统的可解释性和可靠性，以及如何将问答系统应用于实际场景中，如医疗、金融等领域。

与常见的网站搜索相比，智能问答在用户描述自己想要的问题后，不需要提取关键词就可以自行分析搜索，直接获得问题的答案。智能问答通过自动分析问题，针对用户提出的问题筛选出多个候选答案，自动选择准确简短的答案。作为智能问答的核心技术之一，语义分析问答技术在智能问答中起着非常重要的作用。智能问答系统通过知识库中的文字匹配、向量匹配和知识图谱逻辑匹配，来快速较准确的匹配出与用户问题相似的几个匹配结果。[2]当前在智能问答系统研究方向和领域与本课题相

似的有 2022 年中国科学院沈阳计算技术研究所商胜彭的基于语义分析的学涯智能问答系统[2]，从数据上来看，学涯测评领域的数据大多来源于教育网站上，且相关的数据并不是很完备。从进行职业推荐的目的而言，近两年则有少数职业推荐系统的实现。

1.3 研究内容

本课题旨在研究面向职业规划的智能问答系统，该系统可以尝试通过自然语言处理技术，帮助用户解决各类职业规划问题。本文将介绍该系统架构研究，包括系统模块介绍、所需技术知识等方面。同时，本文还将探讨系统的应用前景和发展方向。

首先，本文将探讨该系统所需的技术知识。该系统的实现需要使用多种技术，包括自然语言处理、知识图谱、机器学习等。其中，自然语言处理技术可以帮助系统理解用户的问题，知识图谱可以帮助系统组织和管理职业规划相关的知识，机器学习可以帮助系统不断优化答案的准确性和相关性。

其次，本文将介绍该系统的模块。系统的核心功能是职业问答，用户可以通过输入问题，获取与职业相关的答案。系统会根据用户的问题，自动匹配相关的知识库和数据源，提供最合适的答案。除了职业问答，该系统还包含职业测评功能，通过多种测试量表来综合用户信息及用户画像进行职业类型推荐，及未来发展建议。用户可以通过这些功能，了解自己的技能和兴趣，选择适合自己的职业选择和发展方向。

最后，本文将探讨该系统的应用前景和发展方向。随着人工智能技术的蓬勃发展，在职业规划领域智能问答系统的应用将会越来越广泛，未来该系统可以通过更加智能化的技术，为用户提供更加个性化、精准的职业规划服务。同时，该系统也可以与其他职业规划相关的应用程序进行集成，形成更加完整的职业规划生态系统。

1.4 论文整体结构

本文有如下四个章节：

第一章首先介绍了在如今的时代背景下本课题的研究意义，简要阐述职业规划和智能问答系统在国内外的研究现状，并提出了本文将要研究的相关技术。

第二章介绍了自然语言处理需要使用的 NLP、jieba 库、gensim 库， 知识图谱构建所需的 Neo4j 和 py2neo 库， 职业推荐模块所需的 Scikit-learn 库和 TensorFlow 库等相关技术和概念。

第三章介绍了面向职业生涯规划的智能问答系统的三大模块的整体设计、关键技术点及其实现过程。

第四章对全文进行了概括总结，总结了目前面向职业生涯规划的智能问答系统的研究内容及其功能实现点，又针对当前的不足为后续的研究提出了适当的未来展望。

第二章 面向职业规划的智能问答系统相关技术及知识

2.1 自然语言处理模块相关技术

本节介绍了自然语言处理、jieba 库及 gensim 库。jieba 库用于单词分割和词性标注，而 gensim 库用于创建字典、构建语料库和计算 TF-IDF 分数。这些技术可以应用于广泛的应用程序，包括聊天机器人、情感分析和信息检索。

2.1.1 自然语言处理（NLP）

自然语言处理（NLP）是一门研究人类语言与计算机交互的学科，是计算机科学，人工智能，语言学关注计算机和人类（自然）语言之间的相互作用的领域，研究能实现人与计算机之间用自然语言进行有效通信的各种理论和方法。1

自然语言处理是通过模拟人类语言理解和产生过程，将语言转化为计算机可处理的形式，并利用计算机技术进行语言分析、语义理解、信息提取、自动问答、机器翻译、情感识别等任务的一门学科。它的研究范畴包括形态学、句法、语义、语用、自动语音识别、机器翻译、信息检索、文本分类、文本生成等多个领域。其技术广泛应用于自然语言交互、人机对话系统、智能客服系统、智能语音助手、搜索引擎、金融风控、医疗健康等众多领域。自然语言处理的发展已经走过了几十年的历程，如今随着人工智能技术的不断发展，自然语言处理正成为人工智能领域中发展最为迅猛和具有广泛应用前景的领域之一。

2.1.2 jieba 库

NLP 中最重要的任务之一是词性标注（POS），它涉及将句子中的每个单词标记为其相应的词性，如名词、动词、形容词等。这个任务对于许多 NLP 应用程序非常重要，例如文本分类、情感分析和机器翻译。

系统中自然语言处理模块需要对用户提出的问题进行自然语言理解和语义分析，以便从知识图谱中获取相关知识，为用户提供准确、全面的答案。

为了预处理用于构建职业规划智能问答系统的中文文本，可以使用 Python 中的 jieba 库对输入的文本进行分词和词性标注，以便后续的处理。系统使用 jieba.posseg 库对中文文本进行 POS 标注。该库是一个流行的开源中文文本分词工具，它使用基于规则和统计方法的组合来识别单词及其相应的词性。创建 process_text 函数将一个中文文本字符串作为输入，并返回一个元组列表，其中每个元组包含一个单词及其相应的词性。

jieba.posseg 库使用的算法基于隐马尔可夫模型（HMM），它是一种统计模型，马尔科夫模型有两个基本假设：

1. 齐次马尔科夫假设：马尔科夫链的当前状态之和其前一刻的状态有关，与其它状态无关；对应的概率语言是：

$$p(x_t \mid y_{t-1}x_{t-1} \ldots y_1x_1) = p(x_t \mid x_{t-1}) \quad (2-1)$$

2.观测独立性假设：当前的观测只与该时刻的马尔科夫链相关，与其它观测及状态无关；对应的概率语言是：

$$p(y_t|y_tx_t \cdots y_1x_1) = P(y_t|x_t) \quad (2-2)$$

系统使用一组概率来确定生成观察输出的最可能状态序列：假设系统的潜在状态不是直接可观察的，但从观察到的输出中可以推断出来。在 POS 标注的情况下，状态是词性，输出是句子中的单词。

2.1.3 gensim 库

系统需要对文本进行停用词过滤，以去除一些常见的无意义词汇，例如“的”、“了”、“是”等。引用 gensim 库来使用 corpora.Dictionary 方法创建文本的字典，使用字典构建文本的语料库。

设置一个停用词列表，将词汇从文本中删除，可以减少文本噪声，并提高后续处理效率：

```
dictionary=corpora.Dictionary([filtered])
corpus=[dictionary.doc2bow([tag[0]])fortaginfiltered]
```

[filtered]是一个列表，其中包含从文本中删除停用词后剩余的单词和它们的词性标签。dictionary.doc2bow（）将每个单词映射到其在字典中的 ID，并返回一个包含单词 ID 和出现次数的元组列表。这些元组构成了文本的向量表示，可以用于计算 TF-IDF 分数。

再对文本进行词向量化，以便计算机可以理解和处理它。使用 TF-IDF 模型来将文本转换为向量表示。TF-IDF 模型将文本表示为一个向量，其中每个维度对应于一个单词，其值表示该单词在文本中的重要性，这类模型可以用于许多 NLP 任务。

这里简单介绍一下 TF-IDF 模型：

TF-IDF（Term Frequency-Inverse Document Frequency）是一种用于信息检索和文本挖掘的常用技术。它通过计算一个词在文档中出现的频率和在整个文集中出现的频率来评估一个词的重要性。TF-IDF 可以用于文本分类、信息检索、关键词提取等任务。在自然语言处理中，TF-IDF 是一种常用的特征表示方法，它可以将文本转换为向量表示，从而方便地进行机器学习和数据挖掘。TF-IDF 有两层意思，一层是"词频"（Term Frequency，缩写为 TF），另一层是"逆文档频率"（Inverse Document Frequency，缩写为 IDF）。TF-IDF 算法的计算步骤如下：

第一步，计算词频：

词频就是某个词再文章中的出现次数，但因为文章的长短不尽相同，为了进行不同文章之间的比较，需要将"词频"标准化。

$$词频(TF) = \frac{某个词在文章中的出现次数}{文章总次数} \quad (2-3)$$

第二步，计算逆文档频率：

这时，就需要一个语料库（corpus），用来模拟语言的使用环境。

$$逆文档频率(IDF) = log_{10}\left(\frac{语料库的文档总数}{包含该词的文档数+1}\right) \quad (2-4)$$

一个词出现的次数越多，分母就会越大，逆文档频率就会越小且越接近 0。而为了避免分母为 0（即文档中不包含该词），分母要加 1。Log 则是对得到的数值取对数。

第三步，计算 TF-IDF：

$$TF-IDF = 词频(TF) * 逆文档频率(IDF) \quad (2-5)$$

当有 TF(词频)和 IDF(逆文档频率)后，将这两个词相乘，就能得到一个词的 TF-IDF 的值。某个词在文章中的 TF-IDF 越大，那么一般而言这个词在这篇文章的重要性会越高，所以通过计算文章中各个词的 TF-IDF，由大到小排序，排在最前面的几个词，就是该文章的关键词。[2]可以看到，一个词在文档中的出现次数越多，TF-IDF 值就越高，而该词在整个语言中的出现次越高，TE-IDE 值就越低。

创建 *process_text* 函数接受文本输入并使用 TF-IDF 分数返回文本的向量表示。该函数首先使用 jieba 将文本分割成单词并为每个单词分配词性标签。它删除停用词并创建剩余单词的字典。函数使用字典构建文本的语料库，并计算语料库中每个单词的 TF-IDF 分数。最后，函数使用 TF-IDF 分数返回文本的向量表示。

总的来说，jieba.posseg 库和 HMM 算法是在中文文本上进行 POS 标注的强大工具。通过准确地识别句子中的词性，可以提高许多 NLP 应用程序的性能，并实现更复杂的语言处理能力。

2.2 知识图谱模块

本节主要介绍使用 Neo4j 作为知识图谱数据库，并使用 Python 的 py2neo 库将节点和关系添加到图形数据库中。

2.2.1 知识图谱介绍

知识图谱本质是一种带有语义信息的异构网络拓扑结构。由实体作为网络结构的节点，节点之间的边代表了实体之间的语义关系。[3]这些关系和属性可以存储在一个巨大的语义网络中，其结构和规模具有很高的复杂性。它的建立需要对海量数据进行分析和挖掘，从中提取实体和概念，并且为它们建立语义关系。知识图谱可以用于知识管理、信息检索、智能问答等任务。在自然语言处理中，知识图谱是一种常用的知识表示方法，它可以将知识转换为图形结构，从而方便地进行机器学习和数据挖掘。

而知识图谱更加广泛的被认知的是一个三元组的表示形式，是基于三元组的语义网发展起来的。知识图谱的三元组表示形式即$G = (E, R, S)$，其中$E = \{e_1, e_2, e_3 \cdots, e_{|e|}\}$代表了知识图谱的实体集合；$R = \{r_1, r_2, r_3 \cdots, r_{|r|}\}$是知识图谱中的关系集合；$S \subseteq E \times R \times E$代表知识图谱中的三元组集合。对于 S 的形式，通常可以表示为（*实体*、*关系*、*实体*）即（h，r，t）。其中 h 为头部实体，r 为关系，t 为尾部实体。实体作为知识图谱中的节点数据，是知识图谱的最基本元素，不同实体之间的边代表了实体之间的关系。每个实体可用一个全局唯一确定的 ID 来表示。知识图谱的三元组表示形式可以方便地进行知识表示、存储、查询和推理，是知识图谱的重要组成部分。例如，在职业推荐的知识库中，创建了职位、技能、教育、公司、研究主题、研究人员、大学等节点，并建立了它们之间的关系。这

些节点和关系可以表示为三元组的形式，例如*（职位，需要，技能）*、*（职位，需要，教育）*、*（职位，在，公司）*等。这些节点和关系的创建为知识图谱的构建提供了基础，为后续的知识图谱应用提供了支持。在知识图谱的构建中，需要将实体和关系存储到数据库中，并能够方便地查询和操作这些数据。

成熟的图数据库如 neo4j，Dgraph，JanusGraph，都具备高效的存储和查询功能，对知识图谱的自然语言处理、机器学习等领域提供了很好的技术支持。通过知识图谱的建立和应用，人们可以更加深入地了解世界，发现实体之间的关系，使得人类的智能化服务和决策更加准确和智能化。

2.2.2 Neo4j 库与 py2neo 库

Neo4J 是一种开源无模式并支持事务管理的图形数据库。不同于传统的关系型数据库，Neo4J 使用图数据结构来存储实体和实体间关系数据，而不是使用表来存储。这种存储方式极大地提升多层次关系查询的效率。[6]它提供了一种高效的方式来处理复杂的数据关系，可以轻松地处理数百万个节点和关系。Neo4J 的 web 界面类似 SQL，可以进行像配置、查询、写入等这样的操作，还提供了可视化功能。同时，Neo4J 还提供了一个灵活的查询语言 Cypher，支持复杂的 SQL 语法。Cypher 的设计目的类似 SQL，用户不必编写图形结构的遍历代码，就能对图形数据进行查询。

具备以下几个重要能力：

1. 通过简单的模式匹配来创建、更新、删除节点和关系。

2.通过模式匹配来查询和修改节点和关系的管理索引和约束等。

3.通过模式匹配来增加、删除、修改节点和关系。

4. 对一些关键字进行过滤，从而提高查询效率。

支持多种数据库管理系统在 Python 中，可以使用 py2neo 库来连接 Neo4j 数据库，并将节点和关系添加到数据库中。py2neo 库提供了一种简单而强大的 API，可以轻松地创建节点和关系，并将它们添加到数据库中。这些节点和关系的创建为知识图谱的构建提供了基础，为后续的知识图谱应用提供了支持。

Py2neo 库是 Python 程序访问 Neo4J 数据库的开源工具包，Python 程序可以利用 Py2neo 连接到 Neo4J 数据库中并使用其提供的 Graph 对象来创建节点和关系。在 Py2neo 中使用 Node 对象构建节点，使用 *Relationship()*构建关系，使用 Create 方法或 Merge 方法来创建节点和关系。在 Py2neo 库中也可以使用 run 方法直接运行 Cypher 查询命令来返回查询结果，借助图数据库的索引功能，还可以提升数据的查询效率。[7]它提供了一个简单强大的 Pythonic 的 API，使得在 Python 中使用 Neo4j 变得更加容易，可以轻松地创建节点和关系，并将它们添加到数据库中。

可以使用以下代码连接到 Neo4j 数据库：

$$from\ py2neo\ import\ Graph$$

$$graph = Graph\left(\text{bolt://localhost:7687}, auth = (\text{neo4j,password})\right)$$

使用 Neo4j 作为知识图谱数据库来表示和存储实体之间的复杂关系，提供了一种简单直观的方法来通过 Python 和 Cypher 查询在图形数据库中创建节点和关系。这对于广泛的应用程序非常有用，例如推荐系统、搜索引擎和知识管理系统。

2.3 职业推荐模块

这里研究两种不同的职业推荐模型，分别使用了 scikit-learn 和 TensorFlow 库。旨在根据用户的职业背景和技能，推荐最适合他们的职位。

2.3.1 Scikit-learn 库

Scikit-Learn 是基于 python 的机器学习模块, 基于 BSD 开源许可证。它是一款简单有效的数据挖掘和数据分析工具，它基于 NumPy、SciPy 和 matplotlib 库。Scikit-Learn 的基本功能主要被分为 6 个部分: 分类、回归、聚类、数据降维、模型选择、数据预处理。Scikit-Learn 中的机器学习模型非常丰富, 包括支持向量机、决策树、朴素贝叶斯、K 近邻等, 可以根据问题的类型选择合适的模型。[8]它还提供了一些工具，如模型选择、预处理数据、模型评估和数据可视化等。Scikit-learn 的 API 非常简单，易于使用，因此它是学习机器学习的理想选择。

该库模型使用 TfidfVectorizer 和 cosine_similarity 函数来计算职位描述之间的相似性。

1.TfidfVectorizer 函数

TfidfVectorizer 函数是一种将文本转换为向量的函数，它将每个单词的重要性与它在文本中出现的频率成反比。类似一种文本特征提取工具，将原始文本转化为基于 TF-IDF 的特征矩阵，从而为后面的文本分析应用提供了必要的基础。特别是对于一些需要进行文本相似度计算、主题分析、文本搜索排序等应用来说，TfidfVectorizer 提供了一种有效的转化手段，可以将文本转换为数字化的形式，便于计算机进行更深层次的分析和处理。同时，基于 TF-IDF 的特征矩阵也具有较好的可解释性，可以帮助人们更深入地理解文本特征。因此，TfidfVectorizer 在自然语言处理和机器学习领域中有着广泛的应用和重要的作用。

2.cosine_similarity 函数

cosine_similarity 函数计算两个向量之间的余弦相似度，它是一种衡量两个向量相似性的函数。

基本计算公式为：

$$similarity = cos(\theta) = \frac{A \cdot B}{|A||B|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n}(A_k)^2} \times \sqrt{\sum_{i=1}^{n}(B_i)^2}} \quad (2-6)$$

余弦函数的函数值在-1 到 1 之间，即两个向量余弦相似度的范围是-1 到 1。当两个向量夹角为 0° 时，也就是两个向量重合时，相似度为 1；如果夹角为 180° ，就是两个向量方向相反时，相似度为-1。

用法如下:

$$cosine\_similarity(vector1, vector2)$$

其中，vector1 和 vector2 分别表示两个向量,这里的向量可以是稠密向量或者稀疏向量。如果是稀疏向量，需要使用 scipy 库中的 sparse 函数来将其转化为稠密向量。

最后，使用收集的用户职业背景和技能计算他们与每个职位的相似性，并根据相似性得分对职位进行排序，再返回得分最高的职位描述作为职业推荐的结果。

2.3.2 TensorFlow 库

TensorFlow 是一个由 Google 开发的开源机器学习框架，它可以用于构建各种机器学习模型，如神经网络、卷积神经网络、循环神经网络等。TensorFlow 的主要特点是支持分布式计算、自动求导、动态图和静态图等多种计算模式，同时还提供了丰富的 API 和工具，方便用户进行模型的构建、训练和部署。TensorFlow 不仅仅为实现机器学习或深度学习提供算法接口，同时也是执行机器学习算法或深度学习算法进行计算的框架。在前端可以支持 Python、C++、java 等多种脚本语言进行编程计算或算法接口调用，而在底层的架构是用 C++、Python 等语言编写而成。[9]

TensorFlow 框架可以方便灵活的搭建在不同的操作系统上，比如 Windows 系统，Linux 系统，Android 系统，IOS 系统，服务器，甚至是大规模 GPU 集群上。除了执行机器学习和深度学习算法外，基于 TensorFlow 建立的大规模深度学习模型已经取得了较好的应用，例如语音识别、自然语言处理、计算机视觉、机器人控制、信息提取等。[9]

在 TensorFlow 中，可以使用 *tfnnembedding_lookup* 函数来实现词向量的查找。使用该库 Tokenizer 和 pad_sequences 函数将职位描述转换为数字序列，并使用 LSTM 和 Dense 层来训练模型。

下面是对这些概念的简单介绍：

1.Tokenizer 函数和 pad_sequences 函数

Tokenizer 函数是一个用于文本处理的类，可以将文本转换为数字序列，并将这些数字序列用于训练深度学习模型。可以使用 Tokenizer 类的 fit_on_texts（）来训练 Tokenizer 对象，并使用 texts_to_sequences 方法将职位描述转换为数字序列。使其可以被输入到 LSTM 和 Dense 层中进行训练。

pad_sequences 函数是一个用于序列处理的函数，可以将序列填充到相同的长度，以便其可以被输入到深度学习模型中进行训练。使用 pad_sequences 函数来将职位描述转换为相同长度的数字序列，并将它们输入到 LSTM 和 Dense 层中进行训练。

用法如下：

padded_sequences=tf.keras.preprocessing.sequence.pad_sequences(sequences, axlen=max_length)

其中，sequences 是一个数字序列的列表，max_length 是要填充到的最大长度。padded_sequences 是填充后的数字序列列表。

2. LSTM 层

TensorFlow 中 LSTM 模型（Long Short-Term Memory）是一种基于循环神经网络的深度学习模型，用于处理序列数据，如文本、音频、视频等。LSTM 模型具有记忆功能，可以捕捉时间序列中的长期依赖关系，因此在自然语言处理、语音识别、图像描述等领域得到了广泛应用。

LSTM 包含一个嵌入层、一个 LSTM 层和一个密集层。其中，嵌入层将单词转换为向量，LSTM 层是长短时记忆网络，用于处理序列数据，密集层是输出层，用于输出预测结果。

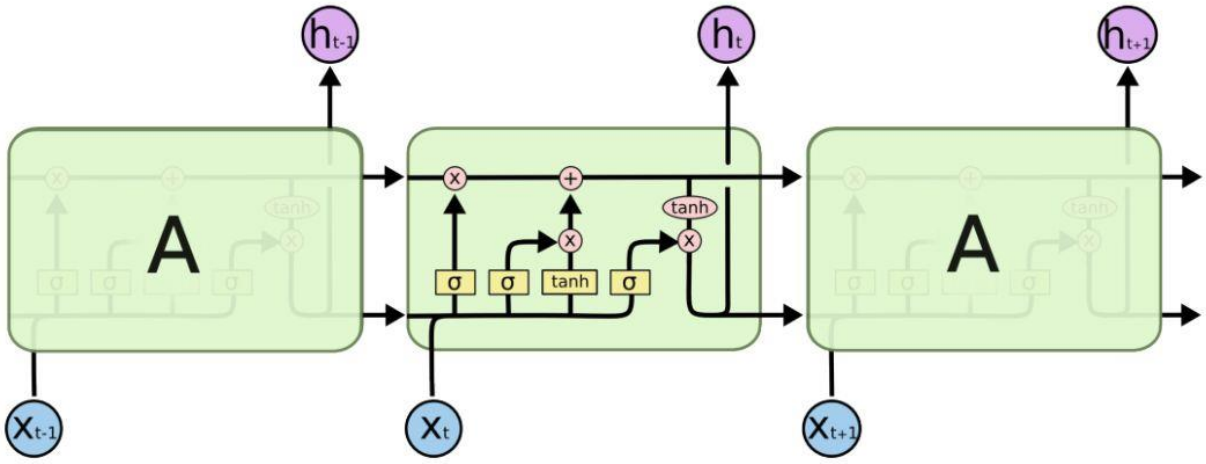

图 1 LSTM 模型结构

LSTM 层包含输入门、遗忘门和输出门三个门控单元，以及一个状态单元。输入门控制新信息的输入，遗忘门控制旧信息的保留，输出门控制输出信息的选择。状态单元是 LSTM 层的记忆单元，用于存储历史信息。

在代码中，使用 LSTM 层来处理数字序列，并将其输入到 Dense 层中进行训练。

LSTM 层的用法如下：

tf.keras.layers.LSTM(units)

其中，units 是 LSTM 层的输出维度。可以将 LSTM 层的输出维度设置为 64。

4.Dense 层

Dense 层是一个全连接层，将离散的词汇映射到低维的连续向量空间中，能够提高模型的表现力和泛化能力。它将输入数据映射到输出数据，并在系统中将 LSTM 的输出转换为最终的输出。在设计代码中，使用 Dense 层来处理 LSTM 层的输出，并将其映射到每个职位的得分。

Dense 层的用法如下：

$$tf.keras.layers.Dense(units, activation)$$

其中，units 是在 Dense 层上的输出维度，activation 则是激活函数。将 Dense 层的输出维度设置为 len(self.job_descriptions)，即职位描述的数量，激活函数设置为 softmax。softmax 函数可以将输出转换为概率分布，以便可以根据概率得分对职位进行排序。

最后，使用收集的用户职业背景和技能计算他们与每个职位的相似性，并根据相似性得分对职位进行排序，再返回得分最高的职位描述作为职业推荐的结果。

2.4 本章小结

文章主要介绍了面向职业规划的智能问答系统相关的技术知识，其中涉及到了自然语言处理模块相关技术，知识图谱技术，职业推荐技术等相关的原理。在进行对比分析之后，可以采取部分更加适合面向职业规划的智能问答系统的相关技术，为后续整体框架的发展做准备。

第三章 面向职业生涯规划的智能问答系统设计与实现

本章结合第二章研究描述的自然语言处理技术、知识图谱构建研究及职业推荐模型的研究，尝试对面向职业规划的智能问答系统框架及功能模块架构进行研究。

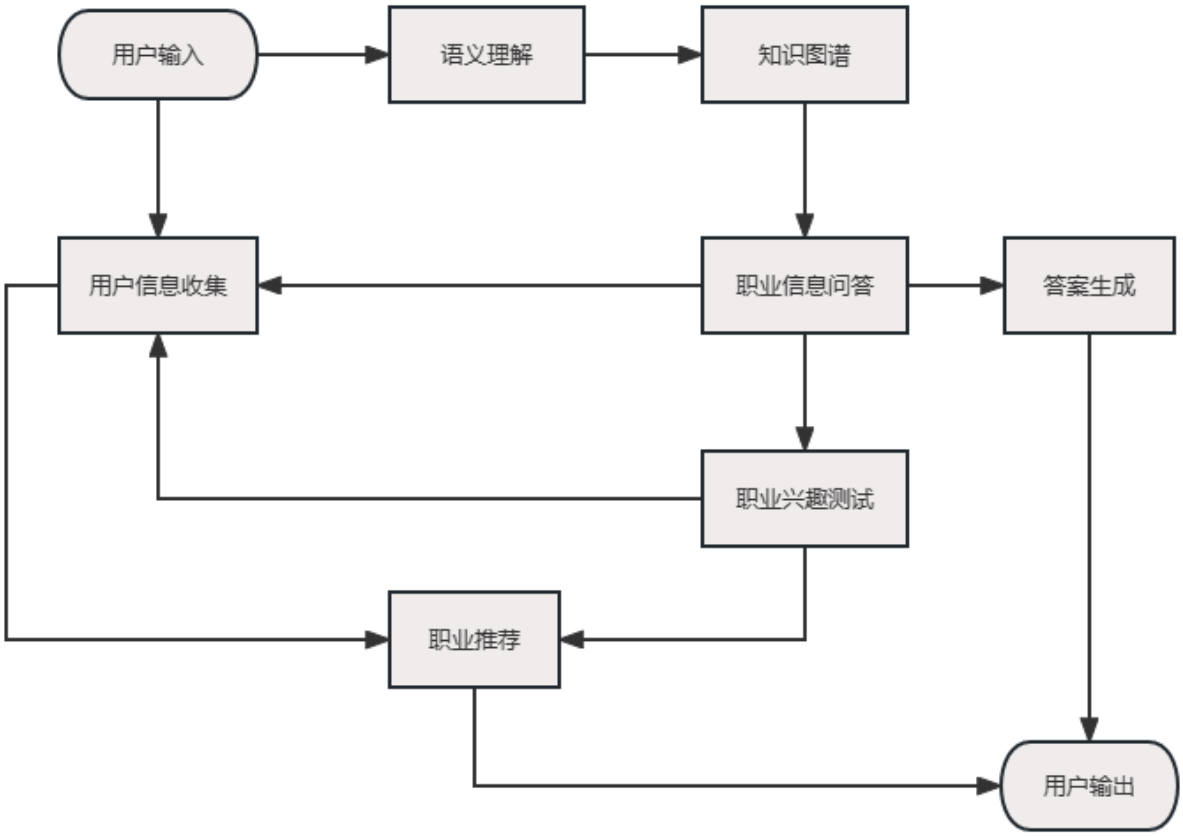

图 2 系统流程图

系统数据流由用户输入开始，分为四个模块，信息收集、职业信息问答、职业兴趣测试、职业发展推荐。在信息收集模块，系统记录并收集用户输入的个人信息到数据库。职业问答模块，系统建立了一个职业信息库，根据用户输入的问题回复用户所需要的信息。职业测试模块，系统输出测试量表的问题，记录用户输入的选项，综合匹配出用户的性格类型及该性格类型的推荐职业。职业推荐模块则是根据用户输入的年龄和薪资，给出初步建议的小工具。

3.1 数据采集和处理模块设计

3.1.1 收集用户信息数据

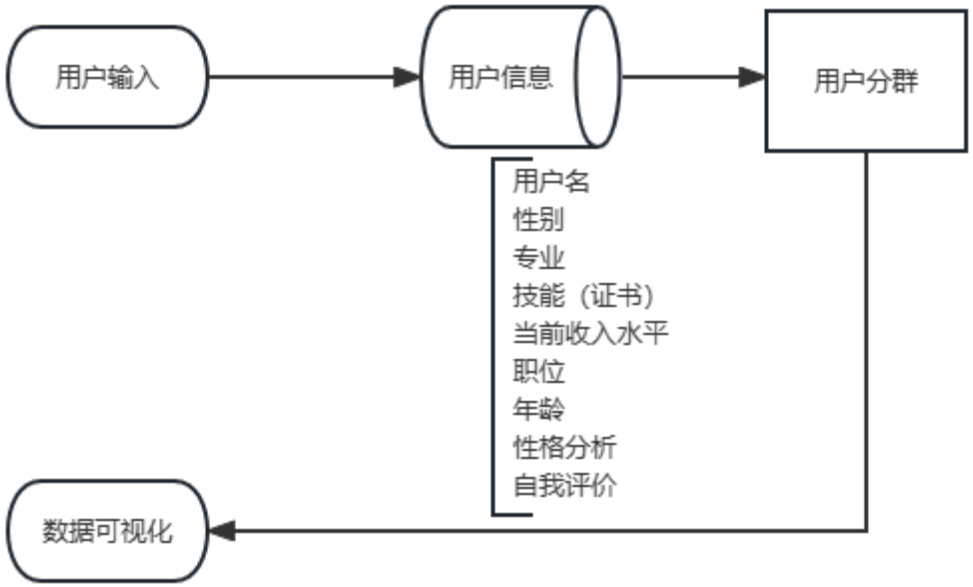

图 3 用户信息数据处理

对这些数据进行处理和分析，以便为用户提供个性化的职业规划建议。目前使用 Python 中的 pandas、numpy 等库对数据进行处理和分析，使用 pandas 读取数据、使用 numpy 进行数值计算。

如果要实现用户画像分析，包括数据加载、数据预处理、特征工程、数据标准化、主成分分析、KMeans 聚类、可视化聚类结果、分析聚类结果、为每个群体推荐职业路径等功能。最终输出了用户画像及职业推荐结果。

本模块的实现可以基于年龄和收入数据对用户进行分群和职业推荐。 首先加载和预处理数据，再通过创建年龄和收入组来进行特征工程。 最后对数据进行标准化，并使用 PCA 将其降到两个维度。对降维后的数据应用 KMeans 聚类将用户分为 5 个群体。 使用散点图可视化聚类结果。 再通过计算每个群体的平均年龄和收入、最小年龄和收入以及最大年龄和收入来分析群体。 基于聚类分析，为每个群体推荐职业路径。 将职业推荐结果添加到数据中，并打印用户的年龄、收入和群体分配以及职业推荐结果。

3.1.2 收集和整理职业相关信息

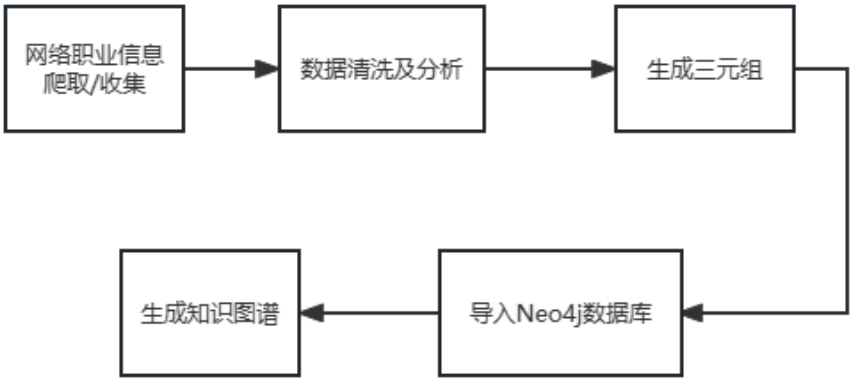

图 4 职业信息知识图谱流程

系统需要建立出一个基于 Neo4j 并使用 py2neo 库来链接的简单知识图谱，便于形成一个用于职业信息查询的知识库。而知识图谱中的实体和关系用一种数学形式来表示，可以使用符号表示或者分布式表示。符号表示是用逻辑符号来表示知识，例如三元组（*实体*1，*关系*，*实体*2），分布式表示是用向量空间来表示知识，例如嵌入向量（$entity1, relation, entity2$）。分布式表示可以利用神经网络来学习知识图谱中实体和关系的嵌入向量，从而捕捉其语义信息和关联性。

使用 py2neo 库来连接 Neo4j 数据库，再用 Node 类来创建节点，用 Relationship 类来创建关系。创建节点和关系后，可以使用 Graph 类将它们添加到数据库中。这些节点和关系的创建为知识图谱的构建提供了基础，为后续的知识图谱应用提供了支持。

表 3-1 知识图谱伪代码

知识图谱创建伪代码具体如下：
1: 创建一个名为 *CareerRecommendationGraph* 的类，该类使用 *GraphDatabase* 驱动程序和 py2neo 库初始化与 Neo4j 数据库的连接。该类提供了在图形数据库中创建节点和关系的方法。
2: 使用 *create_node* 函数创建具有给定标签和属性的节点。*create_node* 方法作为一种静态方法，将事务对象、标签和属性作为输入。构造一个用 Cypher 查询来创建具有给定标签和属性的节点，并使用事务对象执行查询。
3: 使用 *create_relationship* 函数在具有给定关系类型的两个节点之间创建关系。*create_relationship* 方法是一种静态方法，将事务对象、开始节点、结束节点和关系类型作为输入。构造一个 Cypher 查询以在具有给定关系类型的开始节点和结束节点之间创建关系，并使用事务对象执行查询。
4: 使用 *create_nodes_and_relationships* 函数在图形数据库中创建节点和关系。创建多个具有不同标签的节点，如工作、技能、教育、公司、研究主题、研究员和大学。在这些节点之间创建多个关系，例如“需要”、“在”、“发布”和“隶属于”等。

表 3-2 知识图谱实体节点属性示例

实体节点名称	属性 1	属性 2	属性 3	举例
Job	Name	Salary	Location	*name*="软件工程师", *salary*="100000", *location*="旧金山"
Skill	Name	/	/	*name*="Python"
Education	Name	Level	/	*name*="计算机科学", *level*="学士学位"
Company	Name	Location	/	*name*="谷歌", *location*="山景城"
Data_analyst	Name	Salary	Location	*name*="数据分析师", *salary*="90000", *location*="纽约"
Research_topic	Name	/	/	*name*="知识图谱"
Researcher	Name	Field	/	*name*="张三", *field*="计算机科学"
University	Name	Location	/	*name*="清华大学", *location*="北京"

表 3-3 知识图谱关系节点类型示例

关系节点名称	实体 1	关系	实体 2
Job_needs_skill	Job	需要	Skill
Job_needs_education	Job	需要	Education
Job_at_company	Job	在	Company
Data_analyst_published_by_company	company	发布	data_analyst
Researcher_writes_paper	Researcher	输入	Research_topic
Researcher_affiliated_with_university	Researcher	隶属于	University

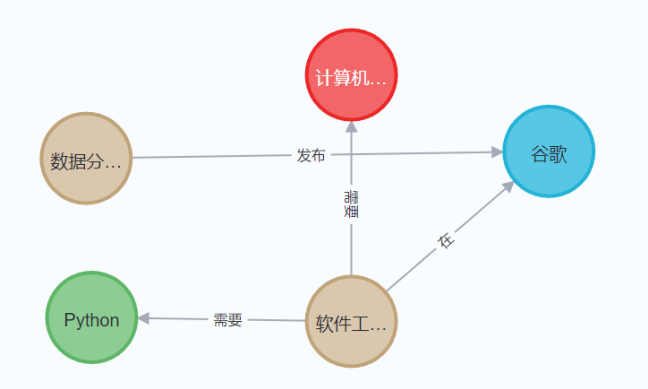

图 5 知识图谱简易示例

知识库的主要数据来自美国劳工部开发的 O*NET online(*the Occupational Information Network*)。知识库的实现使用了 Python 中的 pandas 库，通过读取从 O*NET online 下载的 Excel 文件转化的 csv 文件构建一个职业信息库。职业信息库内的信息有职业类型的定义、职业所需的专业和技能、职业发展的前景、职业所需的受教育程度等等。该信息库可以用于职业分类、职业推荐、职业问答等应用场景。在职业分类中，可以根据职业编码将职业划分到不同的类别中，从而实现职业分类的目的。在职业推荐中，可以根据职业描述和用户的兴趣爱好等信息，推荐与用户兴趣相关的职业。

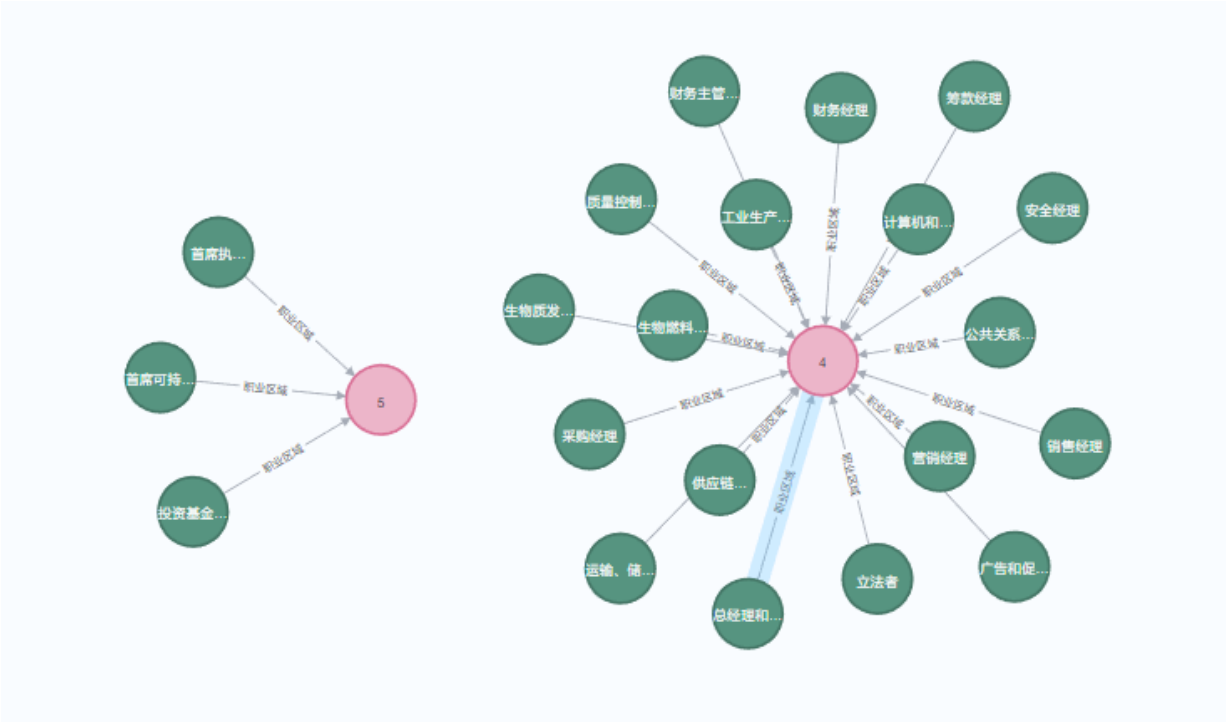

图 6 知识图谱最终结果部分展示

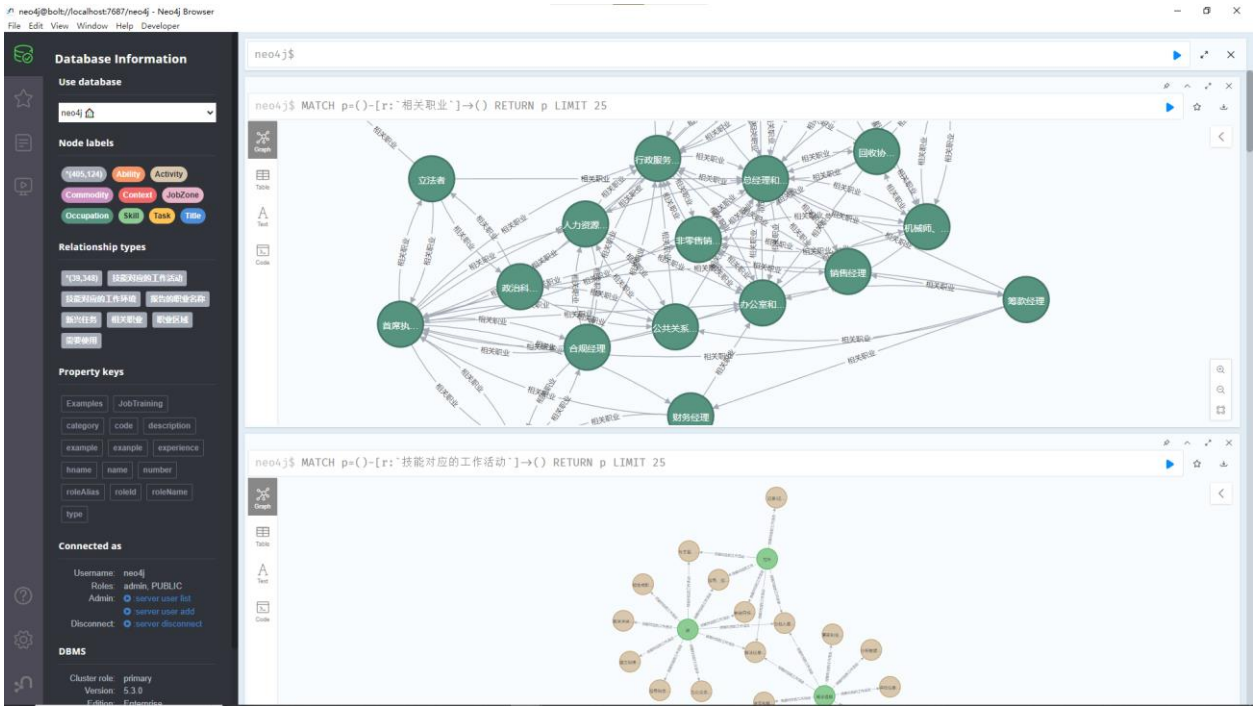

图 7 知识图谱最终界面展示

这一功能模块需要通过不断扩充和更新知识图谱的内容和结构，来提高问答系统的知识覆盖度和准确度。未来也可以通过使用更先进的知识表示和推理技术，例如图神经网络、逻辑推理、符号计算等，来提高问答系统的知识利用率和推理能力。为未来系统的不断完善和发展提供了基础数据支持，为职业规划研究和职业发展方向研究提供了便利。

3.2 职业信息问答模块实现

前面的数据模块负责存储和维护一个包含各种职业、技能、薪资、路径等信息的知识图谱，从而为问答系统提供丰富和准确的知识支持。根据用户输入的文本进行语义理解，使得这一模块可以进行知识检索，从而得到相关的答案，例如“程序员”对应的技能有“编程语言”、“数据结构”、“算法”等。

该模块主要有三个层面：语义理解层、知识图谱层和答案生成层

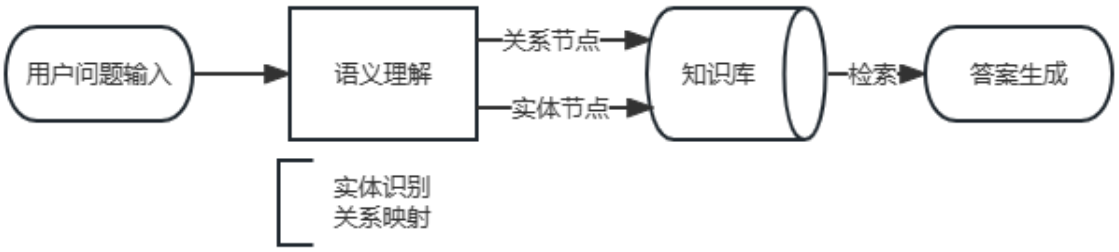

图 8 职业信息问答模块

3.2.1 语义理解层

本层对用户输入的信息进行语义理解

以下是这一块具有的功能及使用的相关技术：

分词：将用户输入的文本分割成一个个的词语，使用基于规则或者基于统计的方法。使用 jieba.posseg 库对中文文本进行分词和 POS 标注。

命名实体识别：从用户输入的文本中识别出一些特定类型的实体，例如人名、地名、机构名等，使用基于规则或者基于统计的方法，例如 jieba.posseg 库使用算法基于的隐马尔可夫模型（HMM）。

意图识别：从用户输入的文本中识别出用户的意图或者目的，例如查询、比较、推荐等，目前使用关键词匹配的方式。

系统实现了一个简单的 NLP 应用，其中包括对输入文本进行分词、去除停用词、计算 TF-IDF 向量以及生成随机回答的功能。这个应用程序可以用于聊天机器人和智能客服系统等场景。首先，process_text 函数接受一个文本作为输入，使用 jieba 库对文本进行分词，并去除停用词。其次，使用 gensim 库中的 corpora.Dictionary 函数将分词后的文本转换为词典，再使用 doc2bow 函数将文本转换为词袋模型。接着，使用 TfidfModel 函数计算 TF-IDF 向量。最后，返回计算得到的向量。generate_response 函数接受一个问题作为输入，并返回一个随机的回答。该函数使用一个包含多个回答的列表，每次调用时随机选择一个回答并返回。这个函数可以用于许多 NLP 应用程序，例如聊天机器人和智能客服系统。

未来系统可以通过引入更多的特征和信息，例如上下文、情感、领域知识等，来提高对用户输入的理解能力和准确度。也可以通过使用更先进的自然语言处理技术和算法，例如预训练模型、神经网络、注意力机制等，来提高对用户输入的理解效率和质量。

3.2.2 知识图谱层

知识图谱的构建使得用户可以通过输入相关问题，获取与职业规划相关的答案。系统将回答与职业规划相关的问题，并获得最合适的答案。

根据语义理解层传递过来的语义信息，在知识图谱中检索出相关的知识，可以使用基于图搜索或者基于向量搜索的方法。基于图搜索是利用图数据库或者图算法，在知识图谱中进行子图匹配或者路径查询，从而找到与语义信息相符合的知识。基于向量搜索是利用向量数据库或者向量索引，在知识图谱中进行相似度计算或者最近邻查询，从而找到与语义信息最相似的知识。本系统采用基于图搜索的方法进行知识信息检索——HNSW（Hierarchical Navigable Small World）算法，通常可以用于高维向量数据的快速近似最近邻搜索。该算法通过构建多层小世界网络，并将向量空间分割成不同的区域，实现了高效的近似最近邻搜索。在系统中应用该算法可以提高搜索效率和准确性。

表 3-4 HNSW 应用方法

具体应用方法包括以下步骤：
5: 将图数据库中的节点向量化，并采用 HNSW 算法构建向量索引。
6: 在问题输入时，使用相同的向量化方法将问题转换为向量。
7: 使用 HNSW 算法在向量索引中查找与问题向量最接近的节点。
8: 根据该节点及其相关节点构造 Cypher 查询语句，并返回查询结果。

知识库中的知识和信息都是事先经过整理和选择的，系统将根据用户的输入，自动检索匹配相关的知识库和数据源。

3.2.3 答案生成层

这一层负责根据用户输入的文本，生成符合用户期望和需求的答案，并将答案转换为文本。

可以采用检索或者生成的方式，检索方式是从预先定义的回答集中挑选出一个合适的回答，生成方式是利用机器学习算法构建深层语义模型，从而生成新的回答。目前系统还处于信息检索阶段，采用检索方式。即用户输入“程序员的技能”，系统输出“编程语言”、“数据结构”、“算法”。

3.2.4 智能问答模块整体实现总结

本系统使用了 Python 编程语言和 Neo4j 图数据库，其中 Python 编程语言提供了 jieba 库，用于对问题进行分词、处理文件和数据格式等操作；而 Neo4j 图数据库则提供了 py2neo 库，用于连接数据库、构造查询语句、执行查询操作等。在本系统中，使用了 jieba 库对问题进行分词，然后构造了一个 Cypher 查询语句，该语句可以在 Neo4j 图数据库中查找与问题相关的信息。

具体而言，首先定义了一个函数 get_answer，该函数的参数为 question，即用户输入的问题。然后，使用 jieba 库对问题进行分词，将分词结果拼接到一个 Cypher 查询语句中，该查询语句可以在 Neo4j 图

数据库中查找与问题相关的信息。最后，执行查询语句并获取结果，将结果格式化为一个字符串，并返回给用户。如果没有找到相关信息，则返回一个提示信息。通过这种方式，可以实现一个简单的智能问答系统，可以帮助用户快速地获取所需的信息。

表 3-5 问答模块伪代码

信息问答模块实现伪代码：
导入所需的库和模块
1: 连接 Neo4j 数据库
2: 定义函数 get_answer(question)：
3: 使用 jieba 库对 question 进行分词
4: 构造 Cypher 查询语句，查询与问题相关的信息
5: 执行查询语句并获取结果
6: 将结果格式化为字符串，并返回给用户
7: 如果没有找到相关信息：则返回一个提示信息
8: 循环读取用户输入的问题，调用 get_answer 函数获取答案并输出:
9: 如果用户输入 exit，则退出循环

问答模块最终实现效果如下图所示：

图 9 系统查询效果

3.3 职业测试模块实现

本模块实现一个根据用户输入的文本（即选择的选项），对用户进行职业兴趣、职业能力、职业适应性等方面的测评，使用基于规则的方法——根据预先定义的测评题库或者测评标准来进行测评，通过一些专业的测评工具，如霍兰德职业兴趣测试量表、MBTI 职业性格测试量表、GABT 测量表等，来评估用户的职业倾向和潜能，记录并综合各类量表的测试结果，以便系统进行职业推荐。

目前使用最基础的算法以问卷形式实现了霍兰德职业兴趣测试和 MBTI 职业性格测试，可以通过综合这两类问卷得出的结果， 进行初步的职业推荐，并记录用户的性格数据和职业倾向。以便后期刻画用户画像，为用户提供更加个性化的服务。

霍兰德职业兴趣测试的直接理论来源是 1959 年约翰.霍兰德教授在长期就业指导实践提出的职业兴趣理论，又称RIASEC理论。它将人的职业兴趣划分为实际型（Realistic）、研究型（Investigative）、艺术（Artistic） 、社会型（Social）、企业型（Enterprising）、常规型（Conventional）六种类型，每种类

型具有相应的特征。本模块实现了Holland职业测试，通过对用户的问答，判断用户的六种职业类型倾向，并给出适合的职业建议。

实现过程如下：

1. 问题分为四个部分，每个部分定义了六个列表，分别对应六种职业类型的问题，每问完一个列表会返回一个整数，表示用户对该类型的倾向程度。

2. 定义了四个 ask_question 函数，分别对应四种问题类型（感兴趣的活动、擅长的活动、喜欢的职业、能力自评），每个函数接收一个问题列表和一个答案列表，遍历问题列表，让用户回答问题，将答案存入答案列表中。

3. 定义了一个字典 Wdict，将六种职业类型的问题列表对应的答案列表存入字典中。

4. 找到 Wdict 中值最大的三个键值对，将键存入一个列表中，作为最终职业倾向。

5. 定义了检查代码，判断列表中的键是否在 career_type 字典中，如果存在，输出对应的职业建议，否则提示用户重新输入。

表 3-6 霍兰德测试模块伪代码

模块实现伪代码：
1: 定义 ask_question1/2/3/4 函数，实现对四种不同类型问题的回答和答案存储。
2: 定义 Wdict 字典，将六种职业类型的问题列表对应的答案列表存入字典中。
3: 找到 Wdict 中值最大的三个键值对，将键存入一个列表中。
4: 判断列表中的键是否在 career_type 字典中
5: 如果存在：输出对应的职业建议，否则提示用户重新输入。

测试情况：

本代码已经通过了多次测试，包括对各种异常输入的处理，以及对各种职业类型的判断和输出。最终实现如下效果。

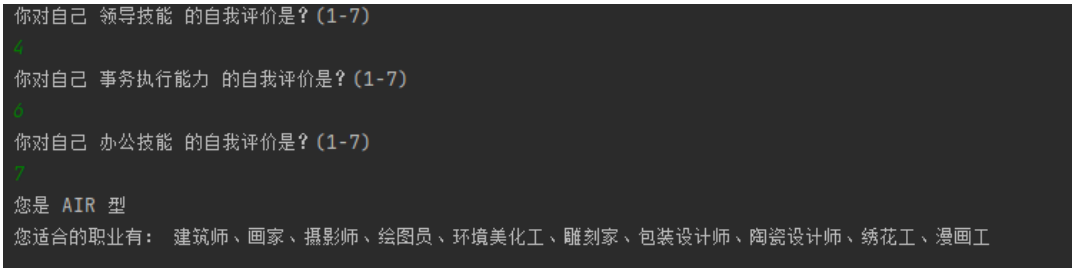

图 10 霍兰德测试结果

迈尔斯-布里格斯类型指标（Myers–Briggs Type Indicator，MBTI）是由美国作家伊莎贝尔·布里格斯·迈尔斯和她的母亲凯瑟琳·库克·布里格斯共同制定的一种人格类型理论模型。

该指标以瑞士心理学家卡尔·荣格划分的 8 种心理类型为基础，从而将人格的心理类型理论付诸实践，经过二十多年的研究后，编制成了迈尔斯-布里格斯类型指标。迈尔斯在荣格的优势功能和劣势

功能、主导功能和从属功能等概念的基础上，进一步提出功能等级等概念，并有效的为每一种类型确定了其功能等级的次序，又提出了类型的终生发展理论，形成四个维度。[10]

本模块实现了一个基于 MBTI 性格测试的职业推荐系统。用户回答一系列问题后，系统会根据用户的回答计算出用户的 MBTI 类型，然后根据 MBTI 类型推荐适合的职业。

表 3-7 MBTI 测试模块伪代码

模块实现伪代码逻辑如下：
1: 定义了两个字典，type_career_XG 和 type_career_zy 分别是 MBTI 类型和性格、职业的对应关系。
2: 定义 questions 作为问题列表，用户需要回答这些问题。
3: 使用 E、I、N、S、F、T、J、P 来分别表示用户回答问题时在不同领域的问题中选择 A 和 B 的次数。
4: 代码使用 E_I、N_S、F_T、J_P 分别表示用户的性格倾向：计算出用户在 E/I、N/S、F/T、J/P 问题中选择 A 和 B 的次数，根据次数的大小关系计算出用户的 E_I、N_S、F_T、J_P 倾向。将这四个倾向拼接起来，得到用户的 MBTI 类型。
5: 定义了 MBTI_type 表示用户的 MBTI 类型。如果用户的 MBTI 类型在 type_career_zy 中，则系统输出用户的 MBTI 类型、性格特点和适合的职业。

测试情况：

在本地运行了代码，并输入了一些测试数据，已经通过了多次测试，包括对各种异常输入的处理。测试结果表明，代码能够正确地计算出用户的 MBTI 类型，并根据 MBTI 类型推荐适合的职业。

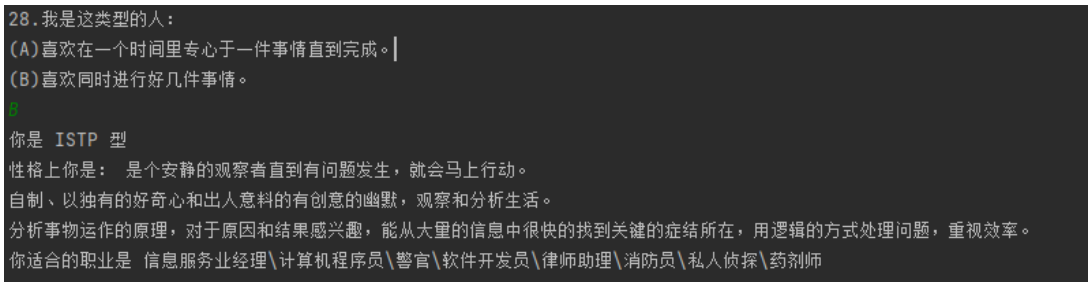

图 11MBTI 测试结果

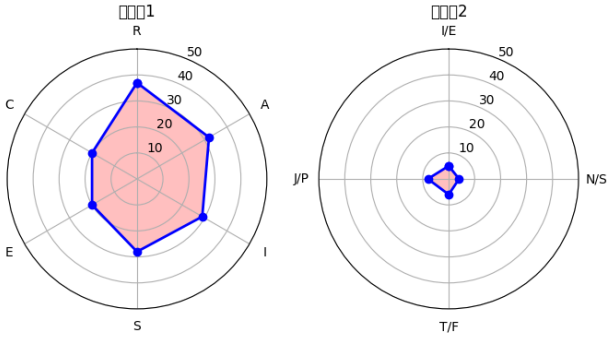

图 12 测试结果可视化示例

最后增加了可视化的部分，这部分定义了两个雷达图函数，分别是 radar_chart1 和 radar_chart2。这两个函数都接受一个列表作为参数，列表中包含了雷达图中每个维度的值。函数中首先定义了雷达图中每个维度的标签和角度，再将角度和数据进行处理，使得雷达图的最后一个点和第一个点重合，从而形成一个闭环。接着，函数创建一个极坐标系的子图，设置了一些属性，如偏移角度、角度方向、极坐标系的最大值等。最后，函数使用 plot 和 fill 函数绘制雷达图，并通过 show 函数显示出来。

未来可以尝试加入更多种量表，更为灵活地为用户了解自身状况和适合的职业发展方向提供便利。

3.4 职业推荐模块实现

本模块实现两个小工具，根据用户输入的文本，对用户进行职业路径和职位推荐等方面提供简单建议的功能。

3.4.1 根据年龄薪资推荐职业发展小工具

用户输入姓名、年龄、性别、公司名称、职位名称和薪资水平，程序根据用户输入的数据，给出职业建议。主要可以使用 pandas 库，用于数据处理和分析。

pandas 是一个开源的 Python 数据分析库，提供了快速、灵活、易用的数据结构，可以用于数据清洗、数据分析、数据可视化等方面。在这一小模块中 pandas 主要用于将用户输入的数据转换为 DataFrame 格式，并进行数据处理和分析。还可以使用 assert 语句进行代码检查，确保输入的数据和处理后的数据符合要求，职业建议正确合理。

本文实现了一个年龄薪资推荐职业的小工具，用户输入姓名、年龄、性别、公司名称、职位名称和薪资水平，程序根据用户输入的数据，给出职业建议。本代码主要使用了 pandas 库，用于数据处理和分析。

表 3-8 年龄薪资推荐职业发展小工具伪代码

模块实现伪代码逻辑如下：
1: 创建 collect_data 函数：收集用户输入的数据，并进行数据类型转换，如果输入的年龄或薪资水平无效，则返回 None。
2: 创建 process_data 函数：处理数据，将用户输入的数据转换为 DataFrame 格式，并根据年龄和薪资水平将数据分为不同的段，分别为“青年”、“中年”、“老年”和“低”、“中”、“高”。
3: 创建 career_advice 函数：根据处理后的数据给出职业建议，根据年龄段和薪资水平段，给出不同的建议。

本模块的算法原理比较简单，主要是根据用户输入的数据，进行简单数据处理和分析，再根据处理后的数据给出职业建议。本代码主要使用了 pandas 库，用于数据处理和分析。pandas 是一个开源的 Python 数据分析库，提供了快速、灵活、易用的数据结构，可以用于数据清洗、数据分析、数据可视化等方面。在本代码中，pandas 主要用于将用户输入的数据转换为 DataFrame 格式，并进行数据处理

和分析。本代码还使用了 assert 语句进行代码检查，确保输入的数据和处理后的数据符合要求，职业建议正确合理。

```
请输入您的姓名：ll
请输入您的年龄：23
请输入您的性别：nv
请输入您的公司名称：l
请输入您的职位名称：l
请输入您的薪资水平：3000
用户输入的数据为： {'姓名': 'll', '年龄': 23, '性别': 'nv', '公司名称': 'l', '职位名称': 'l', '薪资水平': 3000}
处理后的数据为： {'年龄段': '青年', '薪资水平段': '低'}
职业建议为： 建议您考虑跳槽，寻找更好的职业机会。
```

图 13 小工具试用结果

3.4.2 为用户提供个性化的职位推荐小工具

用户提供自己的职业背景和技能，模型就可以推荐最适合他们的职位，这一推荐小工具的实现使用 scikit-learn 和 TensorFlow 库，旨在根据用户的职业背景和技能，推荐最适合他们的职位，通过自然语言处理和深度学习技术来分析职位描述和用户的职业背景和技能，并计算它们之间的相似性。这些模型可以帮助用户更快地找到最适合他们的职位，未来也可以帮助雇主更快地找到最适合他们的候选人。

首先，职业推荐小工具基于 Sikit-learn 库的模型。使用 TfidfVectorizer 和 cosine_similarity 函数来计算职位描述之间的相似性。

其次，通过用户的职业背景和技能计算他们与每个职位的相似性，并根据相似性得分对职位进行排序。

最后，返回得分最高的职位描述。

模块有两种不同的职业推荐模型实现方式：

职业推荐工具定义了两个类：CareerRecommendationModel 和 CareerRecommendationModelTF。CareerRecommendationModel 使用 scikit-learn 的 TfidfVectorizer 和余弦相似度来推荐职位。CareerRecommendationModelTF 使用 TensorFlow 的 Tokenizer、LSTM 和 Dense 层来推荐职位。

CareerRecommendationModel 类的逻辑：

1. 初始化时，将职位描述列表传入，使用 TfidfVectorizer 将职位描述转换为 tf-idf 矩阵，再使用 cosine_similarity 计算相似度矩阵。

2. 推荐职位时，将用户简介传入，使用 TfidfVectorizer 将用户简介转换为 tf-idf 矩阵，再使用 cosine_similarity 计算用户简介与职位描述的相似度，返回相似度最高的职位描述。

CareerRecommendationModelTF 类的逻辑：

1. 初始化时，将职位描述列表传入，使用 Tokenizer 将职位描述转换为序列，再使用 pad_sequences 将序列填充为相同长度，构建 LSTM 模型（如第二章介绍的所示结构）。

2. 训练模型时，将用户简介传入，使用 Tokenizer 将用户简介转换为序列，再使用 pad_sequences 将序列填充为相同长度，使用 categorical_crossentropy 作为损失函数，使用 adam 作为优化器，训练模型。

3. 推荐职位时，将用户简介传入，使用 Tokenizer 将用户简介转换为序列，再使用 pad_sequences 将序列填充为相同长度，使用训练好的模型预测每个职位描述的概率，返回概率最高的职位描述。

```
# 设有以下职位描述和用户简介
job_descriptions = [
    "数据分析师",
    "机器学习工程师",
    "前端开发工程师",
    "后端开发工程师",
]
user_profile = "我是一名数据分析师，熟悉Python和机器学习"
```

图 14 职位描述及用户简介设置示例

```
1/1 [==============================] - 0s 22ms/step
['后端开发工程师', '数据分析师', '机器学习工程师', '前端开发工程师']
['数据分析师', '机器学习工程师', '前端开发工程师', '后端开发工程师']
```

图 15 小工具试用结果

第一行为应用 Scikit-learn 库构建的 CareerRecommendationModel 类运行的结果，第二行为应用 TensorFlow 库构建的 CareerRecommendationModelTF 类运行的结果。

3.5 本章小结

本章主要介绍了对于各模块的功能及其实现的研究，包括数据采集和处理模块、职业测试模块、职业推荐模块、职业信息问答模块等，形成本系统的初级框架。

第四章　总结与展望

4.1 总结

随着社会的发展和经济的变化，互联网上的越来越多的复杂冗长的信息，变化不断的就业市场，越来越多的人需要职业规划，却苦于搜索职业信息的繁杂方式，及缺乏对自身清晰的认知而没有清晰且准确的职业发展方向。在如今高速发展的时代，随着人工智能技术的不断发展，智能问答系统在职业规划领域的应用将会越来越广泛。

本文在对自然语言处理、知识图谱、推荐算法等技术研究的基础上，在进行对比分析之后，尝试对面向职业规划的智能系统的框架和功能模块进行研究，这些功能模块可以为用户提供个性化的职业规划服务功能，如个性化的职业推荐、职业测试和职业信息问答。具体内容有：

（1）在数据采集和处理模块中，可以使用 pandas、numpy 等库对数据进行处理和分析，未来可以尝试实现基于 KMeans 聚类的用户画像分析方法。

（2）在职业测试模块中，可以使用霍兰德职业兴趣测试、MBTI 职业性格测试、GABT 测试等，评估用户的职业倾向和潜能，进行职业推荐。

（3）在职业推荐模块中，可以根据年龄、薪资等因素推荐简单的职业发展规划，也可以根据用户的职业背景和技能推荐最适合的岗位。

（4）在职业信息问答模块中，可以根据用户输入的问题，从知识库中提取与该问题相关的实体和关系，并将这些实体和关系存储到知识图谱中，以便回答与职业规划相关的问题。

4.2 展望

未来面向职业规划的智能推荐系统也将会通过更加智能化的技术成为实现个体和社会共同发展的重要基础。为用户提供全面、准确的职业信息和发展方向，帮助人们更好地规划自己的职业生涯。

在未来的研究中，需要继续：

（1）探索和优化对于用户画像的刻画和用户信息收集的精准度，可以添加社交网络分析与挖掘，实现对用户的职业兴趣、能力和经验进行分析和挖掘，进行聚类分析，并可视化聚类结果，以优化职业推荐算法和模型，提供更为精准更适合用户的职业选择及职业规划。

（2）系统未来优化可以尝试增加 matplotlib、seaborn、sklearn 等库实现一种基于 KMeans 聚类的用户画像分析方法。对基于 KMeans 聚类的用户画像分析方法的研究如下：

a) 加载用户数据，并进行数据预处理。在数据预处理阶段，需要删除缺失值和重复值，并重置索引。再进行特征工程，将年龄和收入分别分组，并进行标准化处理。再尝试进行主成分分析，将数据降维到二维空间。主成分分析将高维数据转化为低维数据，从而更好地展现数据的特征和结构。

b) 使用 KMeans 聚类算法对用户进行分群。KMeans 聚类是一种常用的无监督学习算法，可以将数据分为多个簇，数据相似度较高的放入同一个簇，这样不同簇之间的数据相似度就会

较低。在 KMeans 聚类算法中，将用户分为 N 个簇。KMeans 聚类算法的核心思想是最小化簇内平方和，即将每个数据点与其所属簇的中心点的距离平方和最小化。

c)将聚类结果可视化，并根据聚类结果为每个群体推荐职业路径。职业推荐是一种常用的个性化推荐方法，可以根据用户的特征和需求，为用户推荐最适合的职业。本方法中，可以为每个群体推荐了三种职业，从职业信息库中选择。

该方法可以帮助系统更好地了解用户的特征和需求，为用户提供更加个性化的服务。

（3）丰富职业信息数据库并尝试不断完善改进智能问答模块，以实现更全面的智能问答系统。

（4）未来系统在智能问答模块可以通过引入更多的对话策略和行为，例如主动提问、多轮交互、情感表达等，来提高问答系统的对话灵活度和用户满意度。也可以通过使用更先进的对话管理技术和算法，例如强化学习、生成式对话模型、多任务学习等，来提高问答系统的对话适应性和智能性。例如“如果你想做程序员，你需要学习以下技能：编程语言、数据结构、算法等。”这样用自然流畅的语言进行回答。可以使用以下的技术和算法：

a) 检索：从预先定义的回答集中挑选出一个合适的回答，回答集可以是静态的或者动态的，静态的回答集是固定不变的，例如模板回答、常见问题回答等，动态的回答集是根据知识图谱层传递过来的知识信息实时生成的，例如实体属性回答、关系路径回答等。检索方式可以基于规则或者基于统计，基于规则的方法是根据预先定义的匹配规则或者评分规则来选择回答，基于统计的方法是根据机器学习或者深度学习算法来计算回答与问题之间的相似度或者概率分布来选择回答。

b) 生成：利用机器学习或者深度学习算法构建深层语义模型，从而生成新的回答，生成方式可以是基于知识图谱或者基于文本。基于知识图谱的生成方式是利用知识图谱中的实体和关系作为输入，通过自然语言生成技术，将其转换为自然流畅的文本。基于文本的生成方式是利用用户输入的问题和知识图谱层传递过来的知识信息作为输入，通过自然语言生成技术，将其转换为自然流畅的文本。自然语言生成技术可以使用循环神经网络（RNN）、长短期记忆网络（LSTM）、门控循环单元（GRU）、自注意力网络（SAN）、变压器（Transformer）等。

（5）不断优化系统，用更轻便更流畅的算法完善系统，提供更多的功能模块，进一步拓用户需求和场景，不断提高系统的满意度和准确率。未来该系统也可以与其他职业规划相关的应用程序（如岗位推荐、岗位招聘等）进行集成，形成更加完整的职业规划生态系统。

参考文献

[1] 王训兵,李晓波,王飞.大学生学业生涯规划现状及对策[J].教育与职业,2012(05):73-74.

[2] 商胜彭. 基于语义分析的学涯智能问答系统的设计与实现[D].中国科学院大学(中国科学院沈阳计算技术研究所),2022.

[3] 王越群. 基于知识图谱的深度推荐系统研究[D].吉林大学,2022.

[4] 熊小舟,刘小康,徐滢,罗坤. 基于知识图谱的智能语音识别案例分析[J]. 集成电路应用,2023,40(01):228-229.

[5] 张嘉伟.关于计算机理解自然查询语言的研究[J].信息技术与信息化,2016(04):116-118.

[6] 唐勇. 基于Neo4J的知识图谱管理系统的分析与设计[J]. 办公自动化,2022,27(12):59-61+55.

[7] 桑丽丽,朱晗. 基于Neo4J的人物事件关系知识图谱构建研究[J]. 电脑知识与技术,2022,18(22):18-20.

[8] 杨忆,李建国,葛方振. 基于Scikit-Learn的垃圾短信过滤方法实证研究[J]. 淮北师范大学学报(自然科学版),2016,37(04):39-41.

[9] 齐照辉. 基于TensorFlow的卷积神经网络应用[D].武汉大学,2018.

[10]张桢宁.了解“人性”才更易沟通——MBTI在绩效沟通辅导中的应用[J].人力资源,2009(14):52-55.

[11]郭家旭,董雷. 基于NPU的光纤振动信号数据预处理算法[J]. 电子设计工程,2021,29(20):156-160.

[12]于纪洋. 基于现实信息环境知识图谱的问答系统研究[D].哈尔滨理工大学,2022.

[13]韦炜. 异种通信数据采集与管理系统的研究[D].贵州大学,2009.

[14]陈文颖. 基于情绪分类模型的新中产阶层审美价值观研究[D].上海交通大学,2009.

[15]林莉.人工智能时代背景下自然语言处理技术的发展[J].电子世界,2020(22):24-25.DOI:10.19353/j.cnki.dzsj.2020.22.011.

[16]周璨.人工智能技术在智能问答系统中的应用[J].信息记录材料,2021,22(09):143-145.DOI:10.16009/j.cnki.cn13-1295/tq.2021.09.067.

[17]张凯亮,陈云.基于深度学习的智能语音问答系统[J].信息与电脑(理论版),2022,34(07):104-107.

[18]彭云克,徐勇,李晓宇等.智能问答系统相关知识的研究[J].信息与电脑(理论版),2021,33(20):166-169.

[19]董佳琳,张宇航,徐永康等.基于知识图谱的新冠疫情智能问答系统[J].信息技术与信息化,2021(06):258-261.

[20]刘芳,于斐.面向医疗行业的智能问答系统研究与实现[J].微电子学与计算机,2012,29(11):95-98.DOI:10.19304/j.cnki.issn1000-7180.2012.11.024.

[21]李梦玲. 基于知识图谱的谷物智能问答系统的研究[D].武汉轻工大学,2022.DOI:10.27776/d.cnki.gwhgy.2022.000376.

[22]黄先红.可持续发展视阈下中职生职业生涯规划能力的现状及对策研究[D].四川师范大学,2021.

[23]陈立.基于语义分析的中医体质智能问答系统设计与实现[D].南京理工大学,2021.DOI:10.27241/d.cnki.gnjgu.2021.000473.

[24]王瑛,何启涛.智能问答系统研究[J].电子技术与软件工程,2019,(05):174-175.

[25]郭振文. 基于知识图谱的人工智能领域知识问答系统[D].大连理工大学,2022.DOI:10.26991/d.cnki.gdllu.2022.000141.

[26]苏叶健.基于知识图谱的职业规划推荐系统构建研究[J].企业科技与发展,2022(08):69-72.

[27]王迷莉.基于Python的大学生职业推荐平台设计[J].信息技术与信息化,2021(08):149-152.

[28]张芳容,杨青.知识库问答系统中实体关系抽取方法研究[J].计算机工程与应用,2020,56(11):219-224.

[29]Fensel D, Şimşek U, Angele K, et al. Introduction: what is a knowledge graph?[M]//Knowledge Graphs. Springer, Cham, 2020: 1-10.

[30]Jiang Z, Chi C, Zhan Y. Research on medical question answering system based on knowledge graph[J]. IEEE Access, 2021, 9: 21094-21101.

[31]You B, Liu XR, Li N, et al. Using information content to evaluate semantic similarityon HowNet. Proceedings of 2012 Eighth International Conference on Computational Intelligence and Security. Guangzhou, China. 2013.

[32]Burke R D,Hammond K J,Kulyukin.Question an-swering from frequently asked question files:experi-ences with the FAQ finder system P[J].AI Magazine,1997(18):57-66.