分　类　号：　TP391　　单位代码：10183
研究生学号：　2017534059　　密　　级：　公开

吉 林 大 学

硕士学位论文

(专业学位)

基于知识图谱的计算机领域胜任力研究与应用

Research and Application of Competency in Computer Field Based on Knowledge Graph

作 者 姓 名：王一博
类　　别：工程硕士
领域（方向）：计算机技术
指 导 教 师：徐昊 教授
培 养 单 位：计算机科学与技术学院

2020 年 4 月

基于知识图谱的计算机领域胜任力研究与应用

Research and Application of Competency in Computer Field Based on Knowledge Graph

作 者 姓 名：王一博

领域（方向）：计算机技术

指 导 教 师：徐昊 教授

类 别：工程硕士

答 辩 日 期： 2020 年 5 月 31 日

吉林大学硕士学位论文原创性声明

本人郑重声明：所呈交学位论文，是本人在指导教师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：王一博

日期：　2020 年 5 月 31 日

摘 要

基于知识图谱的计算机领域胜任力研究与应用

随着网络与人工智能的迅猛发展，计算机领域进入高速发展时代，企业对计算机领域人才的招聘更加频繁，高校也越发重视计算机领域的人才培养。但在求职过程中，岗位匹配主要以关键词搜索为主，返回的招聘信息较为分散，岗位所需的知识和技能无法被全面展示，这将影响求职者对于岗位的认知甚至职业选择。为有效解决上述问题，学术界和工业界通过定义岗位胜任力模型为招聘与求职提供客观科学依据。

本文结合计算机领域的特点开展研究，基于求职网站中的招聘数据构建知识图谱，抽取不同类别岗位所需的知识和技能即岗位胜任力，最后研发应用平台提供基于知识图谱的招聘信息语义检索服务并展示全面的岗位胜任力即岗位所需的知识和技能以及所对应的掌握程度。

本文的主要研究和贡献有：

1. 定义了计算机领域招聘信息的知识图谱数据模式和语义关系，包含不同类别的实体、实体间关系、实体的属性等，并在数据模式中引入知识和技能实体，在知识图谱中融合岗位胜任力元素。

2. 构建了计算机领域的知识图谱。收集互联网中计算机领域的招聘信息数据并构造领域词典，使用卷积和双向长短期记忆相融合的神经网络算法抽取出知识图谱中的知识，将整合后的知识存储在 Neo4j 图数据库中。本文的知识抽取方法在实验中具较高的 F1 值，所构建知识图谱具丰富的实体和语义网络。

3. 提出了基于知识图谱的岗位胜任力需求模型抽取方法。本文先基于知识图谱完成对招聘需求文本的语义扩展，基于人工标注和预训练模型使用 BERT 模型将计算机领域招聘信息分为不同类别的岗位。最后使用 word2vec 完成程度词分类，基于共现矩阵实现计算机领域不同类别岗位所需胜任力的抽取，深度挖掘每类岗位所需的知识和技能及相应的掌握程度。实验表明该方法的实验结果较为符合求职的实际情况，对于求职者全面了解岗位具有较好的借鉴意义。

4. 搭建了基于知识图谱的计算机领域胜任力管理平台。本文整合 Neo4j 中知识图谱和岗位胜任力数据，将数据同步至 ElasticSearch。借助 Elasticsearch 的高效优质的全文检索为求职者提供快速丰富的招聘信息语义检索服务，使用 Vue.js 框架搭建具有图谱可视化效果的职位信息列表和岗位胜任力展示平台。

本文所提出知识图谱构建技术和岗位胜任力抽取技术在实验数据中表现良好。基于知识图谱的招聘信息语义检索案例在速度和语义相关度方面具有不错的表现，通过知识图谱可视化展示的岗位胜任力可以较为全面地展示计算机领域不同岗位所需的知识和技能。所搭建的平台可以为求职者提供优质的招聘信息检索和岗位胜任力展示服务。

关键词：

知识图谱，实体识别，岗位胜任力，文本分类，语义检索

Abstract

Research and Application of Competency in Computer Field Based on Knowledge Graph

With the rapid development of the network and artificial intelligence, the computer field has entered a high-speed development era, and enterprises are recruiting more talents in the computer field more frequently, and universities are paying more and more attention to the talent training in the computer field. However, the job matching process is mainly based on keyword search. The returned recruitment information is scattered. The knowledge and skills required for the job cannot be fully displayed, which will affect job applicants' perception of the job and even career choices. In order to effectively solve the above problems, academia and industry provide objective scientific basis for recruitment and job search by defining the post competency model.

In the paper, research will be carried out based on the characteristics of the computer field, and a knowledge graph will be constructed based on the recruitment data on the job site. The knowledge and skills required for different types of jobs will be extracted as post competency. Finally, the research and development application platform provides a semantic search service for recruitment information based on knowledge graph and a comprehensive display of post competency. Demonstrate the knowledge and skills required for the job and the corresponding level of mastery.

The main contribution of this paper can be summarized as follows:

1. The knowledge graph data schema and semantic relationship of recruitment information in the computer field are defined. The data schema includes different types of entities, inter-entity relationships, entity attributes, etc., and introduces knowledge and skill entities into them, trying to incorporate job competence elements in the knowledge graph.

2. A knowledge graph of the computer field is constructed. Crawl the recruitment information data in the computer field on the Internet and construct a domain dictionary.

Use a neural network algorithm that combines convolution and bidirectional long-term and short-term memory to extract the knowledge in the knowledge graph and store the integrated knowledge data in the Neo4j graph database. The experimental results show that the knowledge extraction method in this paper has a higher F1 value, and the constructed knowledge graph has rich entities and semantics.

3. A method for extracting job competency based on knowledge graph is proposed. Extend the semantics of recruitment requirements text based on the knowledge graph. Based on manual annotation and pre-training models, the BERT model is used to classify recruitment information in the computer field into different categories of posts. Then use word2vec to complete the classification of degree words. Finally, based on the co-occurrence matrix, the ability of different categories of posts in the computer field is extracted. Dig deep into the knowledge and skills required for each type of position and the corresponding mastery level. Experiments show that the results of this method are more in line with the actual situation of job hunting, and have good reference significance for job seekers to fully understand the position.

4. A computer-based job competency management platform based on knowledge graph is built. This paper integrates the knowledge graph and job competence data, and synchronizes the data to Elasticsearch. With high-quality full-text search of Elasticsearch, platform provide job seekers with a fast and rich recruitment information semantic retrieval service, and use the Vue.js framework to build a job information list and job competency display platform with graphical visualization.

The knowledge graph construction technology and post competency extraction technology proposed in this paper performed well in experimental data. The case of semantic retrieval of recruitment information based on knowledge graph has a good performance in terms of speed and semantic relevance. The post competency displayed through the visualization of knowledge graph can comprehensively demonstrate the knowledge and skills required for different positions in the computer field. The

established platform can provide high-quality recruitment information retrieval and job competency display for job seekers.

Keywords：

Knowledge Graph, Entity Recognition, Post Competency, Text Classification, Semantic Retrieval

目 录

第 1 章 绪 论

1.1 研究背景及意义

在整个社会信息化的时代背景下，计算机领域的工程专业化信息人才将始终作为信息技术发展的主要推动力[1]。数据显示，在刚刚过去的 2019 年人工智能相关岗位需要约 110 万从业者[2]。面对如此巨大的计算机领域人才缺口，目前高校设置了计算机相关类别专业，为社会提供人才输出。然而在高校毕业生求职过程中出现就业难的情况，这并不是因为每年的毕生数量太多或者水平不足，主要是因为在求职网站中常使用关键词匹配岗位，而这与学生在校期间所学内容无法实现针对性的匹配[3]；另一方面则是企业发布的岗位需求描述中包含的知识和技能较为分散，单个招聘信息无法全面展示岗位所需的胜任力，导致求职者在求职过程中只能片面地了解岗位而影响其职业选择[4]。由此可见，岗位所需的胜任力和人才所具备的胜任力在求职过程中由于结构性和匹配的不合理，可能导致部分求职者无法准确搜索到与之胜任力匹配的岗位，从而影响求职者的就业选择甚至出现就业难的情况。

随着人工智能和大数据的兴起，大批计算机专业毕业生成为计算机领域求职者的主力军，如何帮助计算机专业学生完成生涯管理并且有效展示计算机领域不同岗位所需的胜任力显得尤为重要。计算机领域学生清晰地了解自身胜任力是求职的前提，简单来说就是有明确的自我用户画像，例如哪些是已经熟悉掌握的知识和技能，哪些是自己具备的软实力等等。而企业发布招聘岗位时能精准表述招聘岗位所需胜任力是招聘的首要前提，在招聘过程中明确列出该类岗位所需的知识、技能这样能给求职者更充足的信息，达到招聘到最符合要求人才的目标。为统一求职者和企业在招聘过程中对不同岗位胜任力的定义，学术界和工业界提出构建岗位胜任力模型的方法筛选人才并提升求职的质量[5]。

2012 年谷歌首次提出知识图谱并应用在搜索等应用中，相比传统的搜索技术，知识图谱通过将信息实体相互关联为增强搜索带来利好[6]。然而要构建涵盖存在的所有知识和实体的知识图谱还需要很长一段路要走，但是针对特定领域构建的知识图谱仍具有实用性。由此可见构建计算机领域知识图谱，通过数据整合和知识关联将岗位胜任力的基本要素连接可以形成针对特定岗位的胜任力图谱，

这将为针对具体岗位所应具备的胜任力要求抽象概念化提供新思路。

本文从计算机领域在互联网中的招聘信息入手，以计算机领域企业招聘需求为基础数据源，完成对多源异构数据的加工处理，将胜任力基础要素引入计算机领域知识图谱的数据模型中再构建图谱，并抽取不同类别岗位所需的胜任力，在所搭建的应用平台中提供基于知识图谱的招聘信息语义检索和岗位胜任力可视化展示。本文所构建的平台不同于以往基于关键词的职位搜索，将结合知识图谱的语义网络为求职者提供关联性更强的语义搜索结果，通过抽取计算机领域岗位所需的胜任力更加全面地为求职者展示不同岗位所需的胜任力及其掌握程度，未来将基于知识图谱提供进一步的知识推理、数据挖掘和更多的胜任力管理服务。综上所述，基于知识图谱的计算机领域胜任力需求模型的构建和应用具有一定的研究意义和应用前景。

1.2 研究现状

知识图谱自 2012 年推出，截止目前已取得了极大的进展和成果，已成为认知智能的基础技术，作为一种有效的知识管理工具正强力地推动智能化的发展[7]。知识图谱技术已经在大规模的简单应用场景中取得显著的效果，目前存在很多高质量通用的知识库为搜索服务提供支撑，其中包括 Freebase[8]、Dbpedia[9]等等。近年来，知识图谱的应用场景转变正呈现全新的形势，其应用场景更为繁杂、需求加深到细分领域、所需的专家知识更为密集、数据资源有限等等。

在计算机领域中，知识图谱可用于预测科研方向，探究专业知识关联等等。其数据多来源于论文、专利、期刊等科研出版物，通过构建知识图谱实现知识搜索和知识推理，搭建应用以帮助计算机领域学者掌握前沿的研究热点和研究方法。在 2017 年 AMinner 发布计算机领域专业知识图谱 SCIKG，该知识图谱包含 1 万个知识概念以及概念间的关系，20 万条专家信息以及 50 万篇相关论文[10]。数据集中包括每个专家的职位、隶属机构、研究兴趣以及 AMiner 的链接，每一篇论文都包括标题、作者、摘要、出版机构和年份等数据信息。此外，由上海交通大学 Acemap 团队知识图谱小组发布的学术知识图谱 AceKG，提供了近 2 亿篇学术论文，8 千万条专家信息，涉及 71 万个领域等其他大量的实体，涵盖权威的学术知识，旨在为众多学术科研项目提供数据支撑[11]。知识图谱也用于帮助学生规

划学习路径，通过整合学习资源和学生的行为数据，构建教学知识图谱生成学生的用户画像，提升学习效率[12]。

胜任力的研究可以追溯至 20 世纪 70 年代，其具体的研究主要为如何明确定义胜任力，胜任力可以分为哪些类别和具体类别的详细概念定义。构建胜任力主要是通过访谈、问卷调查和数据统计等方法，针对不同领域和岗位工作内容可以定义不同细分领域的胜任力，常用场景为求职和招聘等[13]。胜任力最初特指业绩较为优秀的人具有的专业和通用知识、职业技能和其他能力要素[14]。随着时间的推移，胜任力的定义被不断补充和完善，胜任力模型可以作为求职、绩效考核、制定培养计划等工作的参考，渐渐成为企业人事部门的衡量指标和管理工具[15]。2017 年 Sergio Miranda 提出基于本体的胜任力模型，在语义网络的层面对胜任力定义又进行了扩展和丰富[16]。

虽然岗位相关胜任力模型在国内的研究时间较短，但是截止目前在中国知网中胜任力相关的文献已将超过 1 万篇。在计算机领域中相关学者通过行为事件访谈和针对具体岗位从业的问卷调查创建计算机领域具体岗位所需的胜任力，其中包括专业知识、行业技能等胜任力要素并将其应用在相关课程的体系设计中[17]。在计算机领域通过构建针对专业人才的胜任力模型来制定人才培养策略，将其引入至计算机教育中以提升学生的学习质量[18]。由此可见在计算机领域关于胜任力的研究较多，但其中多为基于访谈和问卷调查等方式，结合知识图谱抽取岗位所需胜任力的应用案例较少。

1.3 本文主要工作

在学术界，基于胜任力模型来制定计算机领域人才培养模式和专业课程体系的研究正逐步增多，本文将聚焦计算机领域招聘需求，构建基于知识图谱的计算机领域岗位胜任力需求模型，其中包括具体的知识和技能以及相应的掌握程度，旨在为计算机领域求职者提供招聘信息的语义检索和不同岗位所需胜任力的展示服务。本文数据来源于互联网中关于计算机领域的招聘数据，结合胜任力要素定义数据模式和语义关联，抽取招聘信息中的实体和关系，完成岗位分类并抽取不同类别岗位所需的胜任力即知识和技能，最后搭建以知识图谱语义检索和岗位胜任力展示为主要功能的平台应用。

本文的主要工作如下：

第一，本文提出了自顶向下的知识图谱构建方案。在语义层面定义计算机领域招聘信息知识图谱的数据模式，其中包括公司、位置、岗位、技能、知识等概念实体，定义各实体之间的关系。而后预处理 Python 采集的数据，再使用 BiLSTM-CNN-CRF 模型完成知识抽取，将抽取的数据存储至 Neo4j 中，形成一套具有可移植性的针对特定领域招聘信息的知识图谱通用构建方案。

第二，本文基于知识图谱语义扩展建立计算机领域岗位分类模型。首先基于知识图谱完成对招聘需求文本中实体的语义扩展，使用 BERT 模型完成岗位分类任务。将人工标注的招聘数据进行词向量表示后加入线性分类器进行训练，完成将招聘需求数据集分成不同岗位的分类任务，最后以采集的招聘信息作为实验数据，通过模型计算获得与实际相符合的计算机领域岗位分类。

第三，本文构建了计算机领域不同岗位的胜任力需求模型。为完成抽取出不同岗位所需的知识和技能以及掌握程度的任务，首先对每类岗位中用来描述胜任力（知识、技能）的程度词使用 word2vec 计算词语相似度，根据相似度得分完成程度词的分类。再抽取出每类岗位中胜任力与程度词的共现矩阵获得计算机领域不同岗位的胜任力需求模型。

第四，本文搭建了一个基于知识图谱的计算机领域胜任力管理(CSCM)平台。该平台将 Neo4j 中的知识图谱和岗位胜任力需求模型数据整合，而后同步至 Elasticsearch 搜索引擎，提供计算机领域招聘数据的全文搜索服务。基于 Vue.js 前端框架实现搜索结果的列表及图谱展示，通过知识图谱可视化展示计算机领域中不同类别岗位所需的胜任力。

本文提出的构建知识图谱的方法具有可扩展性和移植性，可基于图谱实现数据检索及可视化；基于知识图谱的岗位分类任务可实现对互联网中招聘信息的岗位分类；提出的计算机领域岗位胜任力需求模型的构建方法可以实现对计算机领域不同类别岗位所需的胜任力及掌握程度的抽取；搭建的应用平台可以实现基于知识图谱的语义检索和岗位所需胜任力的可视化。

1.4 本文组织结构

本文以互联网中的计算机领域招聘信息为数据基础，以抽取计算机领域岗位

胜任力为求职者全面展示岗位所需的知识和技能为出发点，以知识图谱为技术切入点开展应用研究，接下来将详细介绍各章节的主要内容。

第 1 章是绪论，首先介绍目前知识图谱和胜任力相关的研究背景、讨论相关研究的实际意义，结合文献阐述目前各方面的研究进展，然后阐述了本文的主要工作及论文的篇幅结构，而后将论文拆分为具体章节详细阐述。

第 2 章是相关研究的概述，分为知识图谱概述、胜任力概述及相关技术概述三方面。知识图谱方面主要介绍本文研究的背景知识，其中包括知识图谱的基础理论知识，主流的构建技术、存储技术和当前主要的应用场景。胜任力概述主要包括国内外的研究现状、模型构建方法和在计算机领域中的应用现状。在相关技术方面主要介绍构建领域图谱的技术架构和关键环节使用的算法。

第 3 章是基于计算机领域招聘信息的知识图谱自顶向下构建方案。首先介绍计算机领域知识图谱数据模式定义，然后针对在招聘需求中的出现的各类实体通过识别算法实现知识、技能等实体及关系的抽取，使用 Neo4j 完成领域知识图谱的存储，通过具体案例展示针对本文图谱的具体操作和存储样例。

第 4 章是计算机领域岗位胜任力需求模型的构建方案。首先对计算机领域招聘需求短文本通过知识图谱完成语义扩展后基于文本分类模型完成岗位分类。再基于 word2vec 对招聘需求文本的程度词分类以便于统一描述岗位胜任力的掌握程度。然后提出基于共现矩阵的胜任力需求模型抽取方法，结合已分类的程度词抽取计算机领域不同岗位所需的知识和技能及其掌握程度。

第 5 章是计算机领域胜任力管理平台的设计与应用。首先介绍平台的总体架构，包括前端、后端及其交互过程，然后介绍平台的关键功能模块，主要为基于 Elasticsearch 的全文检索和基于 Vue.js 的数据展示两模块，最后通过展示平台中的部分前端页面进一步详细介绍平台功能模块的界面及可视化效果。

第 6 章是总结与展望。全面地归纳本文的研究内容和主要贡献，对本文存在的不足之处提出改进方案。然后介绍未来本文的后续工作，包括基于知识图谱的计算机领域胜任力管理相关的研究内容和将要扩展的应用功能。

第 2 章　相关研究概述

2.1 知识图谱概述

关于知识图谱的定义需要追溯到 2012 年谷歌提出知识图谱的概念，谷歌将其应用在搜索中，提升了搜索的性能和效率。狭义的知识图谱特指基于一个或多个领域的多源异构数据构建的知识网络，而广义的知识图谱指大数据时代知识工程中一系列技术的总称，在一定程度上指代大数据知识工程这一新兴的学科[19]。

传统的语义网络中包含概念及其之间的关系[20]，但是概念与实体没有明确的区分，无法对同类语义不同数据进行进一步的表示，其中的边和节点无法实现更加丰富的定义。随着互联网的兴起和工业界数据的激增，知识图谱中的实体、关系和灵活的数据模式更加适用于目前的巨大规模数据库[21]。知识图谱的优点在于语义更为丰富，一方面在设计知识图谱的数据模式时可包含各类语义关系，另一方面语义关系的建模也多种多样。知识图谱的数据具有多源特性，可以通过多个数据来源验证简单的事实，使得数据质量更为精良。知识图谱的结构通常可以表示为典型的图结构，借助 RDF 表示的知识图谱结构更为友好[22]。

知识图谱的研究意义在计算机领域可见一斑。由于其规模足够巨大使得机器易于理解不同的实体和概念，语义关系足够丰富使得机器理解不同的关系，结构友好简化了机器处理程序，数据质量优良让机器对现实世界产生正确的理解。以上满足了实现机器对自然语言理解所需的条件，可以说知识图谱是认知智能的基石。此外知识图谱能更好地实现人工智能，由于图谱中包含的概念、属性和关系的定义，使得机器可以仿照人类的思维解析文字和其中的关系逻辑；知识图谱有助于增强机器学习的能力，整合大量多源异构数据和专家知识的知识库可产生精良的专业领域数据，将之应用在知识增强的机器学习后将得到更完善的结果。目前基于知识图谱技术可以实现精准数据分析、精准搜索意图理解、智能化推荐、知识型推荐等，知识图谱也使得人机交互更加自然，实现一切皆可问答。

2.2 知识图谱构建概述

知识图谱中数据主要以单条知识进行组织，每条知识可以拆分为[主语，谓语，宾语]，这些三元组并不是混乱的堆砌在知识图谱中，而是按照一定的逻辑

存储和表示的[23]。知识图谱的数据模式按照本体论的思想勾勒出数据的组织方式，描述数据之间的相互关系，明确对象的属性，数据模式的分类反映出数据之间的关系特征和数据的内在特征。数据层是根据数据模式组织起来的一个个具体的知识，如果说数据模式是知识图谱的骨架那么数据层就是肌肉，两者构成了一个健壮的整体。在构建知识图谱时，是先确定数据模式再收集具体的数据，还是先收集具体的数据再确定数据模式，形成了两种构建知识图谱的方式[21]。

自顶向下方式首先根据知识图谱的特点设计数据模式，等于首先确定知识图谱的数据收集范围和数据之间的关系和组织方式。这种方式适用于构建行业知识图谱，因为在行业中的数据和组织方式相对确定，组织方式较为清晰。例如构建历史时期人物的知识图谱，可以将人物分类，统计人物的亲属、朋友、敌对等关系，依据这些关系设计数据模式进而收集人物数据形成知识图谱。自底向上的构建方式则是与自顶向下相反，先按照三元组的方式收集数据，然后根据已有的数据提炼数据模型。通常使用这种方式是因为在构建知识图谱初期无法明确数据范围，不清楚数据如何使用，简单的收集所有数据形成庞大的数据集，最后通过整理、分析、归纳、总结形成数据模型。这种方式适用于构建公共领域知识图谱，因为公共领域涉及海量的数据，包含各个方面的知识，内容大而全，这种情况只能先根据数据内容提炼特征形成数据模型。例如谷歌、百度的知识图谱属于典型的公共领域知识图谱，在构建这种图谱过程中，需要不断积累才能对数据知识分类，慢慢呈现出知识架构。知识图谱的两种构建方式在构建初期区别比较明显，但是在构建后期可能会出现相互结合的情况。在自顶向下的构建过程中，随着数据量的不断增加，需要对原有的数据模式进行完善，修订数据模式就类似于自底向上的构建过程。

在构建知识图谱的总体架构中除设计图谱的数据模式之外，还需要引入其他技术框架完成知识抽取和知识融合。其中将抽取的知识要素对应数据模式存入数据层完成知识抽取任务，基于知识抽取结果完成知识融合后生成知识图谱，实现语义检索和图谱可视化等功能。

在图谱构建过程中通常使用自然语言处理的相关技术，从类似表单的半结构化数据与类似文本的非结构化数据中完成属性、关系和实体抽取。通过数据整合、

实体对齐、知识推理完成知识融合，将多源异构的数据进行合并和整合，然后通过质量评估对知识进行筛选和处理，最后将数据存入图谱中，搭建基于已有知识图谱的平台以实现多种功能，构建行业图谱的关键流程如图 2.1 所示。

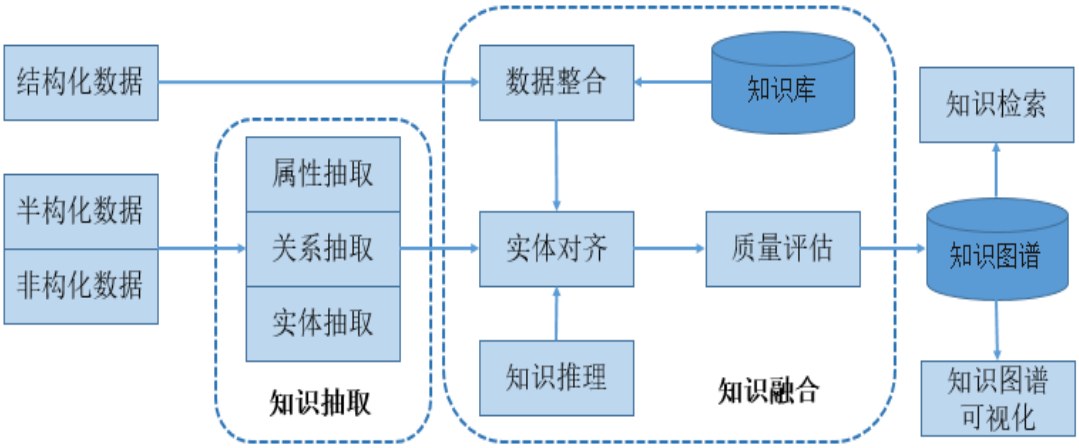

图 2.1 构建行业图谱的关键流程图

2.2.1 实体识别

在自顶向下的知识图谱构建方法中首先进行的是命名实体识别，抽取出在数据模式中定义的实体。知识图谱根据数据源种类的不同所采取的实体识别方式也略有不同[24]。结构化数据主要来源于第三方知识库、网页表单或者各类百科数据，这些数据有一定的标准组织形式，实体识别相对容易，主要是将结构化数据映射为 RDF 形式。非结构化数据的实体识别任务是科研学者关注的重点，通过定位和分类完成非结构化数据中的实体识别任务。目前针对非结构文本数据的实体抽取和分类的方法较多，其中较为典型方法为：基于规则和词典的方法、基于统计的方法。

（1）基于规则和词典的方法

基于规则和词典的方法通常采用语言专家构造的包含标点、关键词、位置词、中心词的模板，通过模式和字符串匹配得到实体[25]。但是由于专家构造的模板对知识库和词典依赖度较高，尤其所定义的规则中常常限于具体的语言、单一的领域和固定的文本风格，导致编制时间较长、移植性不佳，需要频繁更新领域知识库来提升实体识别能力。

（2）基于统计的方法

基于统计的方法需要较高精度的特征词，需要从文本中标注各种特征然后加

入到特征向量中[26]，主要做法为基于训练语料完成数据收集和分析并从中挖掘特征。其中典型方法主要有最大熵马尔科夫模型[27]（MEMMs）、支持向量机[28]（SVM）和条件随机场[29]（CRF）等等。其中最大熵马尔科夫模型的通用性较好，但复杂性较高且需要明确的归一化计算，运算开销比较大。支持向量机主要思想是几何间隔，也可以说是一种硬分类，在完成识别任务时具有相对较高的准确率但其速度相对较慢。条件随机场标注框架具有全局特征灵活的特点，但在完成命名实体识别任务时存在训练时间久和收敛效果不佳的问题。

2.2.2 关系抽取

关系抽取是构建图谱的重要环节，研究者们追求准确高效地在知识图谱中添加更丰富的知识，试图寻找高效自动的关系抽取技术以提升效率。关于关系抽取可以通过一个简单的例子描绘其过程，给定一个句子：“故宫位于北京市”，以及实体“故宫”和“北京”，关系抽取通过语义得到“位于”的关系，抽取出（故宫，位于，北京市）的三元组，即一条知识。在知识图谱的构建过程中，关系抽取具有近二十年的研究历史，目前已有的研究主要分为以下三种方案：

（1）有监督学习：这种方案中监督学习的关系集合比较确定，通常视为一个简单的分类问题。借助机器学习的计算能力完成分类任务，对知识图谱中定义的关系完成二分类，但是实验效果和训练数据有一定的关系，通常来讲数据越优质其实验效果就越好。这种方法的缺点是需要大量人工标注数据，耗时且容易出现人为误差，难以扩展新增其他关系，泛化能力低。

（2）半监督学习：这种方案相较有监督学习则只需要少量的标注信息作为种子，从多源异构的数据中再抽取出大量的新实例构成训练数据，其中典型的方法为远程监督学习[30]。远程监督的思想是如果一条知识在图谱里曾被标记，那么当出现被标记知识中的一对实体时，这个句子中的实体将同样被标记为具有这种关系。这种启发式标注规则有利有弊，在自动标注训练数据时发挥积极作用，但是由于其强势的设定将可能产生错误的标注。近年来，如何排除远程监督数据中的噪音标注干扰成为学者们热议的话题，有研究表明在远程监督学习中引入对抗训练可以提升模型对噪音的抵抗能力[31]。

（3）无监督学习：这种方案通常对语料库中的大量冗余数据做聚类，基于聚类结果定义知识图谱中的关系。这种方法虽然不需要人工标注降低了人为因素影响，但是由于聚类方法本身存在模糊的关系和低频实例召回率低的问题，导致无监督学习很难取得良好的效果。

2.2.3 知识图谱的表示与存储

知识图谱的存储方式可以按照数据规模简要划分，常规知识数据使用 SPO（Subject-Predicate-Object）即包含主谓宾的元组表示，使用关系型数据库可以实现少量的图谱的存储，但当图谱的规模量巨大时，则需选择成熟的图数据库。

（1）基于 RDF 的知识表示：由 W3C 制定的标准数据模型 RDF[32]（Resource Description Framework）常用来表示一条知识。RDF 共有 3 种元素：主体、谓语及客体，每种元素具有统一资源标识符（Uniform Resource Identifier, URI），在整个网络或者图中唯一标记实体或者其他资源，这与日常生活中的身份 ID 类似。例如“天安门广场位于北京市”的文本经过实体抽取和关系抽取，可以组成三元组“天安门广场-位于-北京市”，其中“天安门广场”为这条知识的主体，“位于”为谓语，“北京市”为客体，这个三元组用来描述天安门广场的属性。

目前 RDF 的序列化方式主要有 RDFS/OWL、N-Triples、JSON-LD 等[33]。RDF 初期的表达能力是有限的，比如无法表达面向对象里的类的概念，这是现实中经常遇到的问题但是 RDF 并不支持，于是引入 RDFS 在 RDF 的基础上增加一些词汇，用来定义类、实例、包含关系等具体功能。OWL 区别于 RDFS 在于 OWL 引入布尔算子、数值约束、属性特性等，进一步增强了 RDFS 的表达能力，丰富了知识表示和推理能力。RDF 的数据是以图形式存储，它的查询语言使用的 SPARQL，与 SQL 类似，RDF 的查询返回一条或多条结果，构成当前待查询的 RDF 图的一个子图。

（2）基于图数据库的图谱存储：图数据库的理论基础是图论，即通过图中的节点、与节点相连的边和节点所具有的属性完成对数据源的表示和存储。常见存储系统有 Neo4j[1]、OritentDB[2]等，下面概述 Neo4j 存储系统的理论和相关操作。

[1] https://neo4j.com

[2] https://orientdb.com

Neo4j 是管理非结构化信息数据的高性能高并发数据库，其中的数据被存储在网状存储结构中。在一个图中包含 Nodes（节点）和 Relationship（关系），其中包含 key/value 形式的属性。由于其嵌入式、高性能、轻量级等优势，已经在多个场景中得到应用，其中包括：社交媒体和社交网络图、反欺诈多维关联分析、企业关系图谱等。Neo4j 的数据操作语言为 Cypher（CQL），它可以如 SQL 一样实现图数据库的修改、查询和关联等操作。

2.3 胜任力相关概述

1973 年哈佛大学教授 McClelland 基于前人的研究将胜任力定义为在工作或生活中可以直接影响个人的绩效表现的特质、行为习惯及能力[14]。1993 年 Spencer L.M.和 Spencer S.M.指出胜任力是能够将优秀者和普通人区分开来的特质，主要包括领域知识、动机等[34]。Koeppen K，Hartig J 等在特定领域中定义胜任力为解决某项工作或问题时个人应该具备的认知或技能[35]。

岗位胜任力的定义是在特定领域中针对具体岗位从业人员所应具备的知识、专业技能和其他胜任力的总和，是员工或者应聘者具备的能够完成该工作任务并可以获得优异业绩的能力。具体的岗位胜任力源于对不同领域中一类岗位的工作内容和业绩要求，该类胜任力与岗位类别一一对应。岗位胜任力具有与个人胜任力不同的特性，一是在岗位胜任力中包含从事该岗位所需的知识、技能、以及可以学习到的与岗位密切相关的深层次素质，二是岗位胜任力与岗位所涉及的职业需求和具体的工作内容息息相关，具体的岗位胜任力是基于具体任务和工作内容获得的，三是岗位胜任力是可以测量的，是可以通过事件访谈、专家咨询、问卷调查等使用统计学方法具体化和量化的[36]。

2.3.1 胜任力模型

McClelland 教授在 1973 年提出素质冰山模型中包括知识与技能、特质与动机等，其中胜任力基本要素被分为显性和隐性，正如冰山分为水上可见部分和水下不可见部分，以此将胜任力基本要素拆分。洋葱模型由 Spencer 等人提出，模型借鉴了冰山模型的胜任力因素，将其从内而外进行了重新分类，由于特质和动机难以被外界改变而放在内层，社会角色、价值和自我概念易于被改变而放在中

间层，知识和技能可以从学习和生活中得到积累而放在外层，内层至外层的演变正是冰山模型从隐性到显性的变化。两种经典胜任力模型如图 2.2 所示。

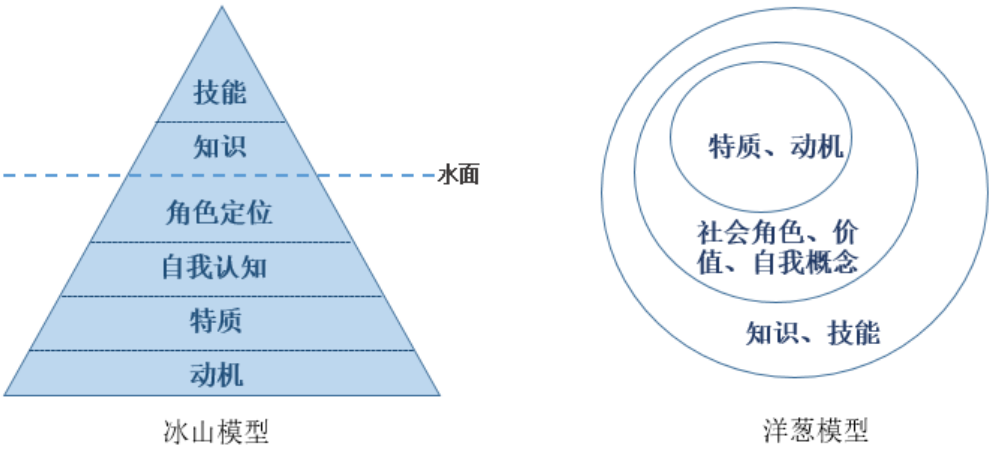

图 2.2 经典胜任力模型

在考虑行业特征时，研究学者主要以某个行业、公司或者从事者这样的固定人群为研究主体，通过类似问卷、访谈等数据采集方式构建胜任力模型。2000 年 MargaretE. Alldredge 针对 3M 公司中的管理人员构建了胜任力模型，其中包括智力水平、道德和判断力等基础要素，还有客户导向、员工培养等必要胜任力[37]。2004 年 Selmer J 等学者针对特定行业的人事管理部门提出人力资源胜任力模型，其中包括财务、业务能力、组织能力、职业沟通等八大要素[38]。在抽取不同岗位所需的胜任力时，明确岗位具体工作职能和需要完成的具体任务是定义岗位所需胜任力的关键，通常基于对该类岗位招聘需求结合不同调研的方法获取具体的工作详细要求后构建模型。

2.3.2 计算机领域岗位胜任力

岗位胜任力的定义针对不同领域都有不尽相同的解读，在任务分配过程中常将模型应用于将工作中特定任务分配给具有相应能力的人，同时也依据岗位胜任力模型来招聘缺少的人才[39]。不同岗位所需胜任力的相关研究在计算机领域也有着广泛的应用，例如基于岗位所需胜任力来探索领域从业者具体的培养方案[40]，或者借助模型全面分析岗位需求，为计算机专业学生制定课程体系[41]等。

岗位胜任力模型是某类领域中与工作相关的知识、技能、特质等素质的综合，具有全面性、多维性、具体性和动态性。计算机领域中专业概念和语义关系随着

时间的推移慢慢增加，其中的关系变得愈加紧密和复杂，在计算机领域的胜任力分析和评判的难度也随之增加[42]。岗位胜任力模型的基本要素包含该类岗位的工作内容和要求，其中岗位的招聘需求描述可以很好的体现岗位的工作内容和技术要求，但是在招聘平台发布简短的招聘需求中涉及各种专业领域词汇、知识和技能以及对以上胜任力特质的掌握程度，不同公司在针对相同岗位的描述也多种多样，并没有形成较为全面的针对具体岗位的胜任力模型。

目前在计算机领域可以应用语义网络和本体模型，通过自然语言处理和语义网络将多源异构数据清洗为可用的数据和关系网络，通过知识图谱进行概念层级的匹配和分析，构建对专业领域知识和技能的胜任力模型[43]，通过系统地数据处理并结合人工适当干预将生成相对科学合理的计算机领域岗位胜任力主线。

2.4 相关技术概述

2.4.1 条件随机场

条件随机场（Conditional Random Field，CRF）满足最大熵原则其本质是一种无向图模型，在科研过程中常用于解决词法分析和知识抽取等标注问题[29]。在 CRF 模型中的输出是没有明确边界的软输出，以标注问题为例，输入全局特征集 X 以及待标记序列 Y，其输出不是明确的结果而是计算 Y 可能的概率即$\mathrm{P(Y|X)}$。在 CRF 模型中不再遵循隐马尔可夫的观测独立假设，也就是除去了图中的方向，在该无方向连通图 G=(V，E)中满足如下条件：

$$\mathrm{P}(Y_v \mid X，\ Y_w, w \neq v) = P(Y_v \mid X，\ Y_w, w \sim v) \quad \cdots\cdots\cdots\cdots\cdots\cdots (2.1)$$

若对任意节点v等式 2.1 成立，则称条件概率分布$\mathrm{P(Y|X)}$是条件随机场，$w \sim v$表示在图 G 里与节点v有关的所有节点w，$w \neq v$表示为在图 G 里与 v 不相同的其它节点w，Y_v与Y_w为节点v，w对应的随机变量。

在完成类似词法分析这样的标注任务时，常用的解决方案是使用线性链条件随机场(Linear Chain Conditional Random Fields, Linear-CRF)完成该类任务，设$\mathrm{X} = (X_1, X_2, \ldots, X_n)$，$\mathrm{Y} = (Y_1, Y_2, \ldots, Y_n)$为线性链模型中的变量，其中$(X_1, X_2, \ldots, X_n)$和$(Y_1, Y_2, \ldots, Y_n)$通过随机方式生成。该模型中在给定 X 取值后，Y 的概率分布以$\mathrm{P(Y|X)}$作为表示，若能够满足下式中的等式，即是线性链条件随机场。

$$P(Y_i|X,Y_1,\dots,Y_{i-1},Y_{i+1}\dots,Y_n)=P(Y_i|X,Y_{i-1},Y_{i+1})\ \ i=1,2,\dots,n \quad (2.2)$$

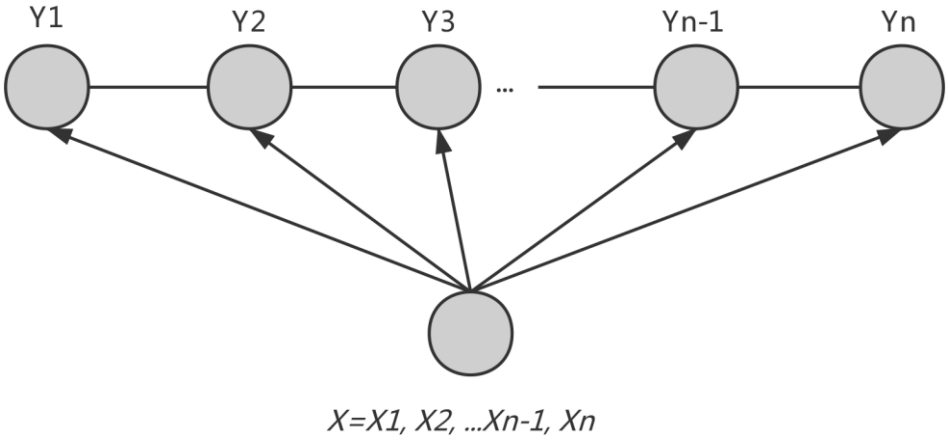

图 2.3 线性链条件随机场结构图

Hammersley-Clifford 相关研究成果表明，在 Linear-CRF 中$P(y|x)$能够被拆分成比邻节点在数学意义中的函数乘积，即$P(Y|X)$可改写为新形式计算方程式，其中X被分解为若干节点x，Y 被分解为若干节点 y，最终条件概率具有如下形式：

$$P(y|x)=\frac{1}{Z(x)}\exp\left(\sum_{i,k}v_k t_k(y_{i-1},y_i,x,i)+\sum_{i,l}u_l s_l(y_i,x,i)\right) \quad (2.3)$$

其中，

$$Z(x)=\sum_y \exp\left(\sum_{i,k}v_k t_k(y_{i-1},y_i,x,i)+\sum_{i,l}u_l s_l(y_i,x,i)\right) \quad (2.4)$$

t_k和s_l是特征函数，其取值为 1 或 0 取决于函数的条件是否得到满足；v_k和u_l是对应的权值，$Z(x)$是规范化因子。

2.4.2 双向长短期记忆网络

长短期记忆网络（Long Short-Term Memory）模型适用于对文本数据的时序数据建模，使用 LSTM 模型可以获取较长距离的依赖关系，通过训练后可以学习到记忆和遗忘的部分信息[44]。LSTM 模型由时刻 t 对应的词X_t，细胞状态C_t，临时细胞状态$\widetilde{C_t}$，隐藏层h_t以及遗忘层f_t，记忆层i_t，输出层O_t组成。通过对细胞状态遗忘旧信息和记忆新信息实现后续时刻的有用信息的传递和无用信息的遗忘，通过 h_{t-1}和X_t计算f_t、i_t、O_t来控制下一个时间步 t 输出的h_t状态。

双向长短期记忆网络（Bidirectional Long Short-Term Memory）由向前方向的记忆网络和向后方向的记忆网络结合而成。由于 BiLSTM 是融合了两组相反

学习方向的 LSTM 层，所以模型能在理论上实现当前词条既包含前文信息又包含后文信息特性，这样将更有利于对当前词条的标注，BiLSTM 结构如图 2.4 所示。

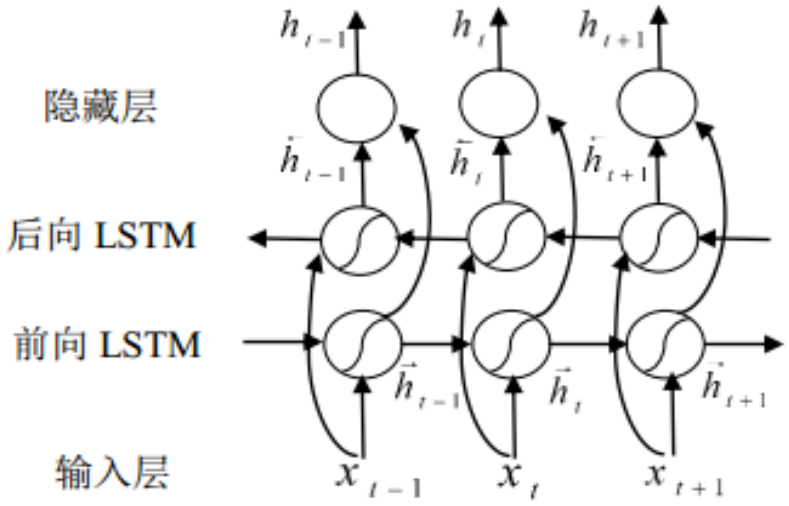

图 2.4 BiLSTM 网络结构图

通过大量的已标注数据和模型不断的迭代优化，BiLSTM 在理论上可以取得不错的效果，但是由于只考虑了标签中的上下文因素，忽略了当前位置的标签和前后位置的标签也可能存在潜在的关系，可能导致无法将这种关系加入到模型中。因此自然语言处理领域的学者提出在 BiLSTM 模型后增加一层 CRF，用于学习获得最优标签序列[45]。整合后的命名实体识别过程可以简单理解为自然语言的句子经过 BiLSTM 进行特征提取输出特征，将特征和对应的标签输入 CRF 后获得自然语言文本中的实体概率序列，具体的网络结构如图 2.5 所示。

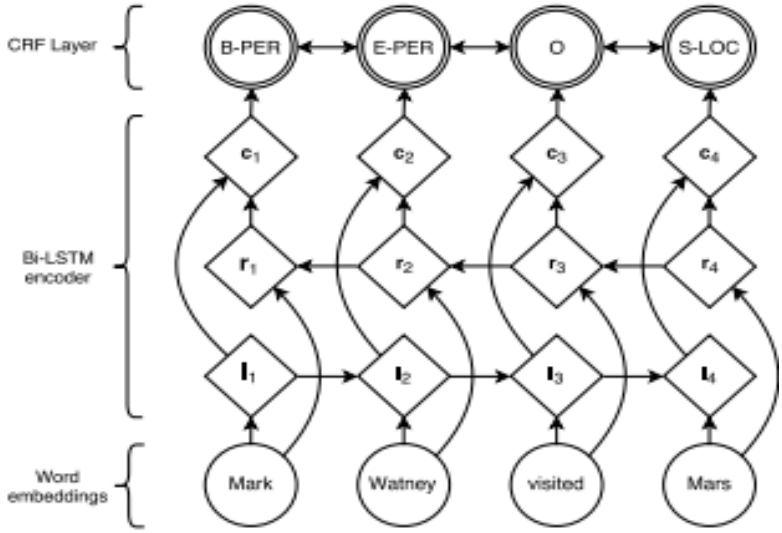

图 2.5 BiLSTM-CRF 模型网络结构图

2.4.3 BERT 模型

BERT（Bidirectional Encoder Representations from Transformer）模型由多层 Transformer Encoder 组装而成[46]，模型结构如图 2.6 所示。BERT 模型的目标是利用大规模无标注语料训练获得文本中包含语义信息的文本语义表示。在模型中为了增强注意力机制的多样性，将每个字的多个增强语义向量线性组合，获得与原始字向量长度相同的增强语义向量。BERT 模型在 Transformer Encoder 结构中增加了残差连接使得网络更容易被训练，增加了 Layer Normalization 对神经网络节点标准化，对每个字的增强语义向量做两次线性变换增强模型的表达能力。最终 BERT 模型的主要输入是文本中的字/词的原始词向量，输出的是在文本中各个字/词融合了全文语义信息后的向量表示。

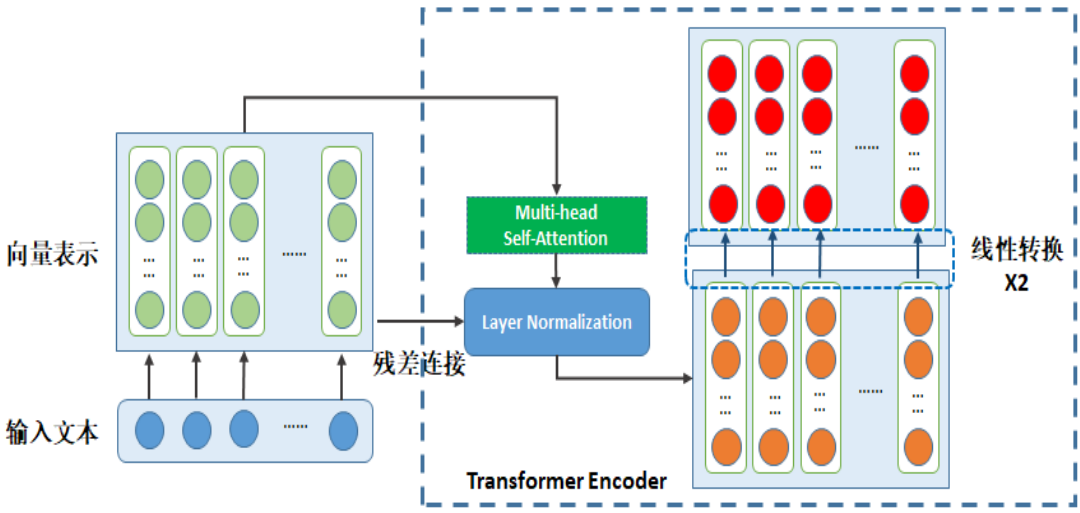

图 2.6 Transformer Encoder 结构图

BERT 模型的输入既可以是单个文本也可以是明确表示文本的句子对，输入文本经过 3 层 Embedding 完成文本嵌入表示，在标记嵌入表示(Token Embedding)和位置嵌入表示(Position Embedding)外还加入了分割嵌入表示(Segment Embedding)用于区别两个句子完成分类任务，具体结构如图 2.7 所示。BERT 模型为达到双向深度上下文学习的效果，采用了两种无监督预测任务进行预训练。一种无监督预测任务是随机遮蔽（mask）掉一个句子中的词，利用上下文进行预测；另一种是用从语料库中简单生成的二进制语句来训练模型，多用于判断句子对之间的关系，典型的应用场景有如问答系统、自然语言推理等。

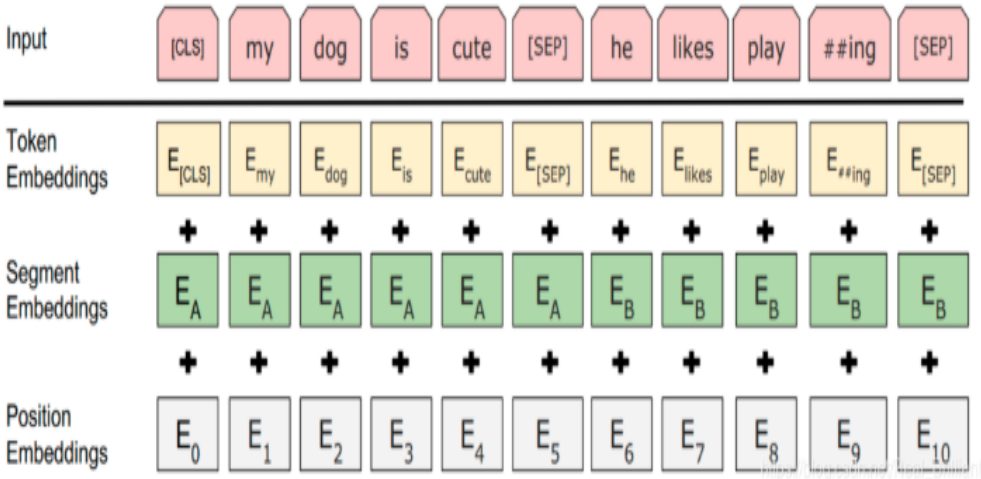

图 2.7 BERT 模型三层嵌入表示结构图

为了使机器能够实现在阅读专业领域文本时实现相关知识的推理，研究学者提出了一种基于知识图谱的语言表示模型(K-BERT)[47]，K-BERT 引入领域知识三元组即引入专业领域知识对文本进行扩展，增加语义信息以提升处理专业领域自然语言处理任务的性能，同时降低大规模训练的成本。

2.4.4 word2vec 词向量模型

word2vec 作为一种自然语言处理工具，可以完成文本的向量化表示、可以度量词与词之间相似度与语义关联[48]。在 word2vec 出现之前，深度神经网络(Deep Neural Networks，DNN）经常被用来完成自然语言处理任务，譬如研究词语间的关系等。DNN 模型一般包含输入(Input)层、映射(Projection)层与输出(Output)层，较为典型的为 CBOW(Continuous Bag-of-Words)与 Skip-Gram 两种模型。

由于 DNN 模型存在词表维度多和计算量大的问题，word2vec 从数据处理方式和数据结构方面对 DNN 模型进行优化[49]，在输入层到映射层方面没有继续采用线性变换与激活函数结合方法而是采用对输入词向量取平均的方法。word2vec 改进了 softmax 计算方式，采用了哈夫曼树作为映射层和输出层的媒介，softmax 概率计算只需沿着哈夫曼树形结构进行即可。和以往的深度学习框架相比，哈夫曼树中的节点相当于神经元，根节点的词向量相当于投影后的词向量，所有叶子节点相当于输出层的神经元，叶子节点的总数就是词汇表的规模，这种优化又被称为分层 softmax（Hierarchical Softmax）[50]，采用分层 softmax 优化使时间复杂度从$\log V$降低为$\log_2 V$。

接下来概述 woord2vec 中基于分层 softmax 优化的 CBOW 模型。CBOW 模模型的输入是上下文词语的初始随机词向量，映射即对其进行向量加法，输出可能为上下文词语中空缺的词语。在 word2vec 中输出采用二元逻辑回归方法存储在树形结构中，二分类决策输出哈夫曼树编码 1 或哈夫曼树编码 0，分别代表向下左转或向下右转。经过编码后所有的全局词语被由 0 和 1 组成的数字串唯一标识，在哈夫曼树中的不同分支路径则代表在输入文本中的不同事件，同样也由 0 和 1 组成的编码唯一标识。经过使用哈夫曼树对数据存储和表示重新定义后，就可以计算得出当前词典中的词在文本中的条件概率$\mathrm{p}\left(w\middle|Context(w)\right)$和对数似然函数$\mathcal{L}$，具体公式如下：

$$\mathcal{L}=\sum_{w\in C}\log p(w|Context(w)) \quad (2.5)$$

$$\mathrm{p}\left(w\middle|Context(w)\right)=\prod_{j=2}^{l^w}p(d_j^w|X_w,\theta_{j-1}^w) \quad (2.6)$$

式中：p^w表示从原始节点出发至 w 对应的路径；l^w表示路径中包含节点的个数；$p_1^w,p_2^w,\dots p_{l^w}^w$表示$p^w$中的各节点；$d_1^w,d_2^w,d_3^w,\dots d_{l^w}^w\in\{0,1\}$代表 w 的由 0 和 1 组成的唯一标识码，$d_j^w$表示$p^w$的 j 节点的路径编码；$\theta_1^w,\theta_2^w,\theta_3^w,\dots\theta_{l^w-1}^w\epsilon\mathbb{R}^m$表示与路径$p^w$对应的参数向量。

2.5 本章小结

本章主要阐述了构建领域图谱的关键环节，基于参考文献介绍了胜任力的相关概念定义，分模块概述了本文所涉及的技术和模型。具体来说本章首先介绍了知识图谱的由来、图谱相较于语义网络的优势和归纳总结目前知识图谱的应用及意义。然后介绍构建图谱的主要技术，从实体识别、关系抽取和图谱的表示和存储几个重要环节阐述技术要点和方案。随后主要围绕胜任力的定义及其相关模型、与计算机领域岗位相关的胜任力定义、主要应用和研究现状展开介绍。最后梳理了构建知识图谱的主要算法和与本文自然语言处理任务相关的模型，下一章将详细介绍计算机领域的知识图谱构建。

第 3 章　计算机领域知识图谱的构建

本文知识图谱的数据来源为互联网中计算机领域的招聘信息，由于其具有较为统一的数据格式，故按照自顶向下的方式构建图谱即先定义计算机领域招聘信息图谱的数据模式。调研互联网中的招聘信息依据高质量的招聘数据和国内研究成果设计数据模式，在其中融入胜任力要素知识和技能，明确在知识图谱中的实体、属性和实体间的关系。接着介绍数据预处理、数据标注和数据采集的流程。最后介绍基于词嵌入的文本向量化、知识抽取和知识存储。

3.1 知识图谱的构建框架

本章构建的计算机领域知识图谱的数据源为互联网中的招聘信息，首先定义知识图谱中的数据模式，其中包括招聘信息数据中不同类别的实体、关系及其详细属性。然后使用 Python 语言编写爬虫实现数据获取，对获取的多源异构数据进行数据预处理同时构建领域词典。最后通过知识抽取模型获得在招聘需求文本中的知识、技能等实体后存储在图数据库中。正如图 3.1 所示，构建过程主要由以下步骤组成。

（1）知识图谱的数据模式定义。经过分析与总结已有的计算机领域知识图谱数据模式设计本文数据模式，其中包括公司、岗位、地点、薪资、技术领域、工具、算法和掌握程度等概念实体以及实体间的关系，定义实体数据类别和属性。

（2）数据获取。编写 Python 程序获取互联网中计算机领域的招聘信息数据，获取的多元异构数据中包含如公司、薪资、地点等结构化数据和招聘需求描述这样的非结构化数据。爬虫程序获取 json 格式数据后经过数据处理将数据解析为在数据模式中定义的概念类别，部分高质量数据则直接存储为相应实体。

（3）数据预处理。将获取数据中部分高质量结构化数据与从计算机领域论文中抽取出的关键词通过解析程序构成计算机领域词典。针对步骤(2)中获取的数据中的文本采用人工标注的方法在其中标注出数据模式中包含的各类实体，已标注好的数据集将作为知识抽取的训练集和测试集。

（4）知识抽取与存储。使用知识抽取算法在步骤（3）的标注数据中完成模型的训练，将其余未被标注的数据输入知识抽取模型中完成知识的抽取。将实体与关系进行数据整合后存入 Neo4j 中完成知识的存储。

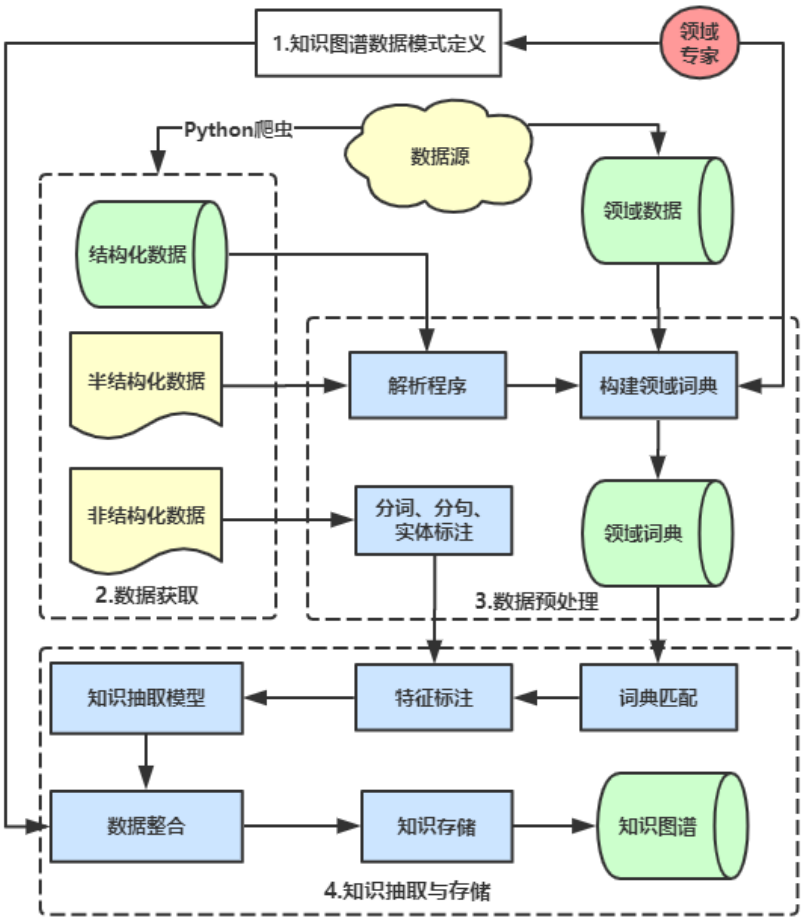

图 3.1 知识图谱构建过程图

3.2 数据模式设计

无论是开放域的知识图谱还是细分领域的行业知识图谱，都需要获取大量的数据，这些数据的选择标准就是数据模式，数据模式主要解决知识图谱的数据组织方式问题。数据模式作为图谱数据的底层架构，其本质是一种知识体系框架，数据模式能够涵盖知识图谱中所有类型的数据。

本文是针对计算机领域基于互联网中的招聘信息数据而构建的行业知识图谱。在招聘信息中定义公司、岗位、地点等实体，还有招聘信息数据的岗位需求描述文本中涉及的针对具体岗位所需的技术领域、掌握程度、工具、算法、编程语言等实体。由于招聘岗位与岗位需求描述是一一对应的关系，所以将掌握程度作为岗位与岗位需求中的工具、算法、编程语言实体间的关系。如此可构建（实体，关系，实体）样式的三元组。本文中使用常用的本体设计工具 protégé 绘制数据模式，使用 owl 本体描绘语言存储并表示数据模式。

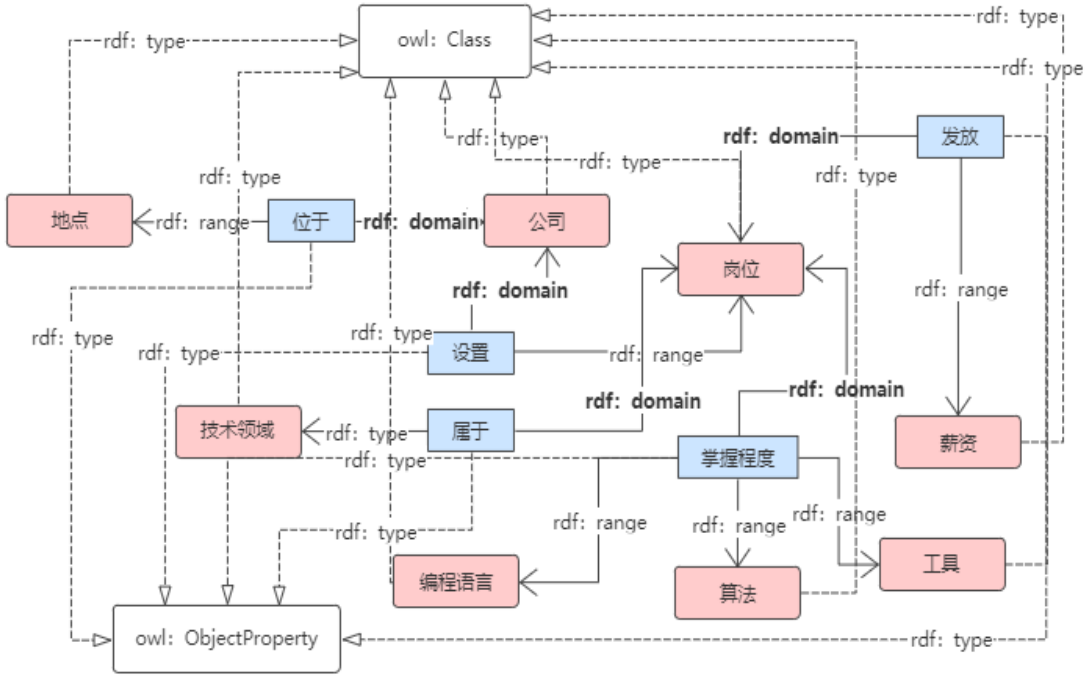

图 3.2 计算机领域图谱的数据模式

图 3.2 表示为包括 8 种实体和 5 种关系的数据模式，其中 rdf：type 连接到的 owl：Class 和 owl：ObjectProperty 表示本体中的类别，相当于图谱概念中实体类别和关系类别；rdf：domain 和 rdf：range 分别表示关系所连接的主语实体和宾语实体。例如由 ObjectProperty：“位于”连接的 rdf：domain：“公司”和 rdf：range：“地点”表示一条知识（公司，位于，地点）。图 3.2 中所有的节点是在本体定义中的概念，在知识图谱中则被替换为实体，例如（公司，位于，地点）的知识可能有（百度，位于，北京）和（阿里巴巴，位于，杭州）等不同三元组。

在数据模式中，不仅包括前文介绍的组织形式，还包括组织形式中每个实体的属性，每个属性都有数据类型和值。在计算机领域招聘信息图谱中，最重要的是岗位这一实体，由图 3.2 数据模式可见，与岗位连接的共计 4 种关系和 6 种实体，表 3.1 列出了岗位的部分属性。在属性列表中也可以存在实体对应实体的情况，表 3.1 中“Organization”就是岗位实体与公司实体之间对应，同时还设置有实体唯一的“编号”属性用于标识唯一资源。列表中的“编号 Id”、“类型 Type”、“名称 Name”、“描述 Description”等为通用类型属性，将被引入到知识图谱中的其他实体中。

表 3.1 岗位实体属性列表

属性名称	数据类型	描述
Id	STRING	唯一标示实体的编号
Type	Class	实体所属的类型
Name	STRING	实体的名称
Description	STRING	对实体的文本描述
Start-time	Moment	开始存在的时间
End-time	Moment	不复存在的时间
Organization	[]Organization	实体所在公司或者其他机构的实体

3.3 数据获取与预处理

3.3.1 数据获取

本文数据来源为计算机领域的互联网招聘信息，接下来以互联网招聘平台拉勾网为例介绍数据获取的过程。拉勾网为每家注册公司分配了一个数字，每家公司子页面使用了较为复杂的 JavaScript 代码和框架，这种架构不适合采用抓包解析 HTTP 协议的方案，经过分析和调研最终决定采用 Python 中的 Selenium 的 WebDriver 驱动浏览器方法模拟人工点击的方式获取数据。在 WebDriver 的关键流程分为三部分：一是 Python 自动化代码用来发送请求给浏览器的驱动；二是浏览器的驱动用来解析自动化测试的 Python 代码，解析后将指令发送给浏览器；三是浏览器用来执行驱动发来的指令，最终完成点击和浏览的操作。本文使用的浏览器为谷歌浏览器和对应版本的浏览器驱动。浏览器通过 HTTP 标准通信协议与服务器实现数据请求与应答，大部分编程语言提供了很多与 HTTP 相关的类库，这样方便数据传输与通讯。WebDriver 获取数据的内容主要来源于 HTTP 协议中的 body 部分，具体包含获取数据中的招聘信息数据和网页表单中的其他数据。本文编写 Python 程序以自动化模拟人为点击网页的方式获取招聘网站中的数据，并通过 JSON 格式的数据来实现数据的传输和解析，将数据分类存储至在本地并生成数据文件。在本文的 Python 自动化脚本中定义了包含初始化设置、遍历解析职位数据、模拟点击翻页以及职位信息写入等函数，具体如图 3.3 所示。

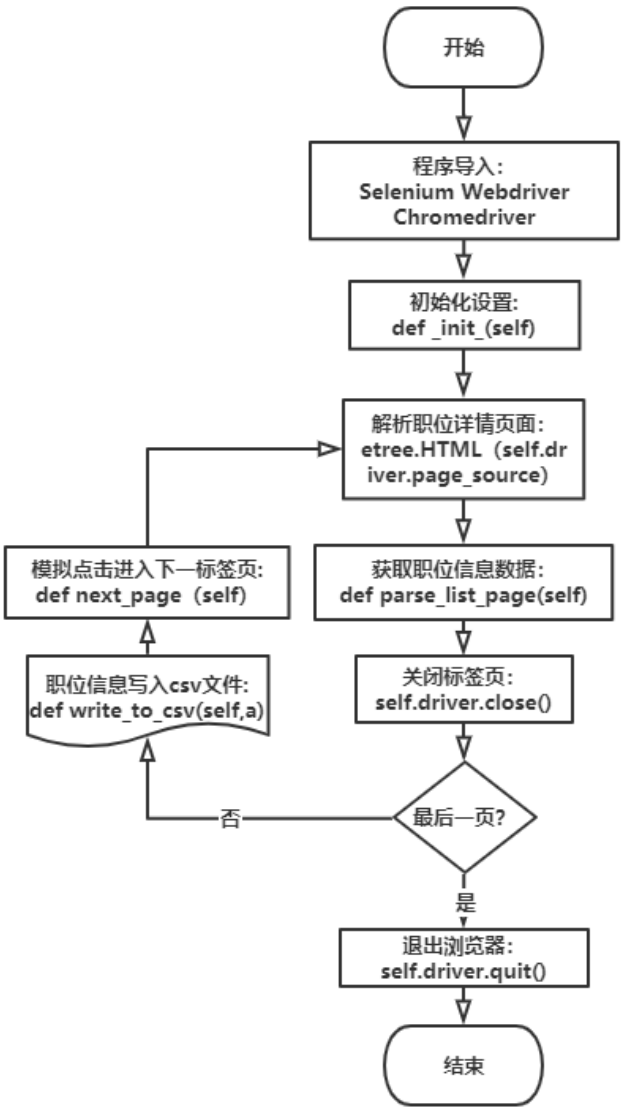

图 3.3 Python 爬虫流程图

通过 Python 程序在互联网招聘网站中模拟人为操作收集数据，总计获取 2500 余条招聘信息，其中涉及 33 个城市的 100 余家公司。

3.3.2 构建计算机领域词典

为了在多源异构数据中更加准确全面地抽取实体，本文构建领域词典为在非结构数据源中抽取实体提供先验知识。本文的数据来源为招聘信息，在数据模式中与“岗位”连接的“技术领域”、“编程语言”、“算法”、“工具”实体需要在招聘需求描述文本中抽取，这些数据仅仅通过人工标注来获取是不现实的，所以需要在人工标注前从专业领域抽取概念来构建领域词典。

在爬取的招聘需求中有关于招聘岗位的标签数据，其中包含较多的领域专业词汇可以作为计算机领域词典的一部分，但是对于门类众多的计算机领域数据显

然是不够的。考虑到在计算机领域中前沿的研究和成果会通过论文和其他出版物发表，所以本文依托计算机领域的前沿论文，对其文本完成分词处理后抽取出其中的专业词汇对计算机领域词典进行扩充。本文分别抽取论文的名称、摘要和关键词，使用 Python 的 jieba 中文分词组件完成分词并去除部分无关描述，剩余的名词和动词视为计算机领域岗位所需技术描绘词汇。本节共收集 500 余篇计算机领域前沿论文，共计抽取 900 余个计算机领域专业词汇。为了更加贴近招聘情境，本文将招聘信息中岗位和公司标签数据与论文的专业词汇融合后构造领域词典，总计 1254 个领域词汇，图 3.4 展示领域词典中的部分词汇。

图 3.4 领域词典词云图

3.3.3 数据标注

本文采集的数据中的招聘需求描述文本属于非结构数据，在知识抽取之前需要对其中的实体类别进行标注便于训练模型并对未标注的非结构化文本完成知识抽取。在爬取的数据中先提取部分招聘需求文本作为标注集，设计不同的标识符在标注集中标注出在数据模式中的各类实体。由于本文采集的招聘信息中存在部分在网页表单中的结构化数据，非结构数据主要为招聘数据中的职位需求描述文本，所以本节在职位需求描述文本中标记编程语言、算法、工具以及对以上实体的掌握程度，人工标注后作为知识抽取的训练数据和测试数据。

标注的数据分为领域词典和 500 条招聘需求描述文本，从学术论文中获得领域词典的词汇较多为技术领域、算法和工具，从招聘需求中的标注则较为复杂，在标注时设计使用不同标识符代表不同实体，“/d#”即 degree 代表“掌握程度”；“/a#”即 algorithm 代表“算法”；“/t#”即 tool 代表“工具”；“/f#”即 field 代表“技术领域”；“/p#”即 programing 代表“编程语言”；“/o#”即 other 代表“与标注无关的数据”等，标注结果如表 3.2 所示。

表 3.2 数据标注列表

实体名称	标注次数
degree-/d#-掌握程度	1924 次
algorithm-/a#-算法	642 次
tool-/t#-工具	1076 次
field-/f#-技术领域	6526 次
programing-/p#-编程语言	1041 次
other-/o#-与标注无关的数据	10691 次

3.4 知识抽取与存储

3.4.1 基于词嵌入的文本向量化

在知识抽取前，文本数据需要经过预处理成为数值张量才能输入到神经网络，这一过程称为文本向量化。通常的文本向量化可以分为将每个单词转换为向量、将字符转换为向量和提取单词或字符的 n-gram 并将其转化为向量三种。本文将输入的数据的每一个中文文字作为文本的标记，标记过程使用 python 程序自动完成分词和标记操作。

在文本标注中，常见的标注体系包括 IO、BIO、BMEWO、BMEWO+等，标注体系越复杂准确率越高，但是相应的标注成本也会增加。结合在招聘需求描述文本数据特点，本文选择 BMEWO[51]完成标注任务，其中 B、M、E 分别为数据模式中实体的开头、中部和结尾，W 为单独实体，O 为无关数据。例如“/f#”人工标注中用来表示一个技术领域的实体，只是表示在数据模式中的一个类别，但是通过分词和标记后单个实体将被重新拆分，用“/B_f”“/M_f”和“/E_f”分别代表技术

领域实体中的每一字的位置和类别，示例如表 3.3 所示。

表 3.3 分词标注对比表

分词前	2./o#熟悉/d#中文的/o#自然语言处理/f#和传统算法/o#
分词后	2/o ./o 熟/B_d 悉/E_d 中/o 文/o 的/o 自/B_ f 然/M_f 语/M_f 言/M_f 处/M_f 理/E_f 和/o 传/o 统/o 算/o 法/o

词向量化的表示方式主要有两种，分别为独热编码(one-hot)和词嵌入(word embedding）。例如“自然语言处理”这句话包括 6 个字，“自”的 one-hot 表示为:[1,0,0,0,0,0]。由此可见 one-hot 表示法和词汇表的大小有直接的关系，假设词汇表中包含 10000 个单词，那么每个单词就需要用维度 10000 的向量表示，得到一张高维度稀疏的张量。而词嵌入则是将单词嵌入到一个低维的稠密空间中，例如“自然语言处理”这句话中的“自”的词嵌入表示可能变成[0.2,0.4,…]，空间维度与词汇表规模呈正相关。对比来看 One-hot 编码简单，但空间维度很高，词嵌入空间维度低且经过训练可以让空间拥有结构。

本文采用 Keras 提供的嵌入层处理输入文本的向量化，Keras Embedding 的优势在于它既可以单独用来学习词嵌入又可以在其他模型中复用。相比于与基于上下文信息的无监督方式 word2vec，Keras Embedding 则是一种基于标签的监督学习。本文中对 Keras Embedding 中的设置参数有：词汇表（input_dim）即中文字符的种类设置为 1000，词向量维度（output_dim）设置为 64，输入数据长度为(input_length)设置为 30。Keras Embedding 层最终将输出一个二维向量，文本中的每一个字对应一个输出序列，这样低维稠密的向量将作为知识抽取的数值张量输入神经网络。

3.4.2 基于 BiLSTM-CNN-CRF 的知识抽取算法

常用基于深度学习的实体抽取模型为 BiLSTM-CRF，其中 BiLSTM 融合两组学习方向相反的 LSTM 层，通过大量的已标注数据和模型不断迭代可获得良好的分词模型。由于 BiLSTM 模型只考虑了标记（Token）的上下文信息，没有考虑到标记的局部信息，在学习长句时可能因为模型容量问题而失去重要信息。因此本文在 BiLSTM-CRF 框架中加入卷积神经网络（CNN）用以记录标记的局部信息。

本文选取一层卷积和一层池化组成 CNN 层结构，将输出结果与 BiLSTM 层输出的字符集向量矩阵相拼接作为全连接层（fully connected layers，FC）[52]的输入。全连接层将文本特征映射到样本标记空间中，再将特征整合到一起输出一个值，减少特征位置对 CRF 分类带来的影响。本文的 BiLSTM-CNN-CRF 的框架中 BiLSTM 层和 CNN 层分别提取标记的全局和局部特征信息，将拼接后的向量输入全连接层后再输入 CRF 层进行解码，具体流程如图 3.5 所示。

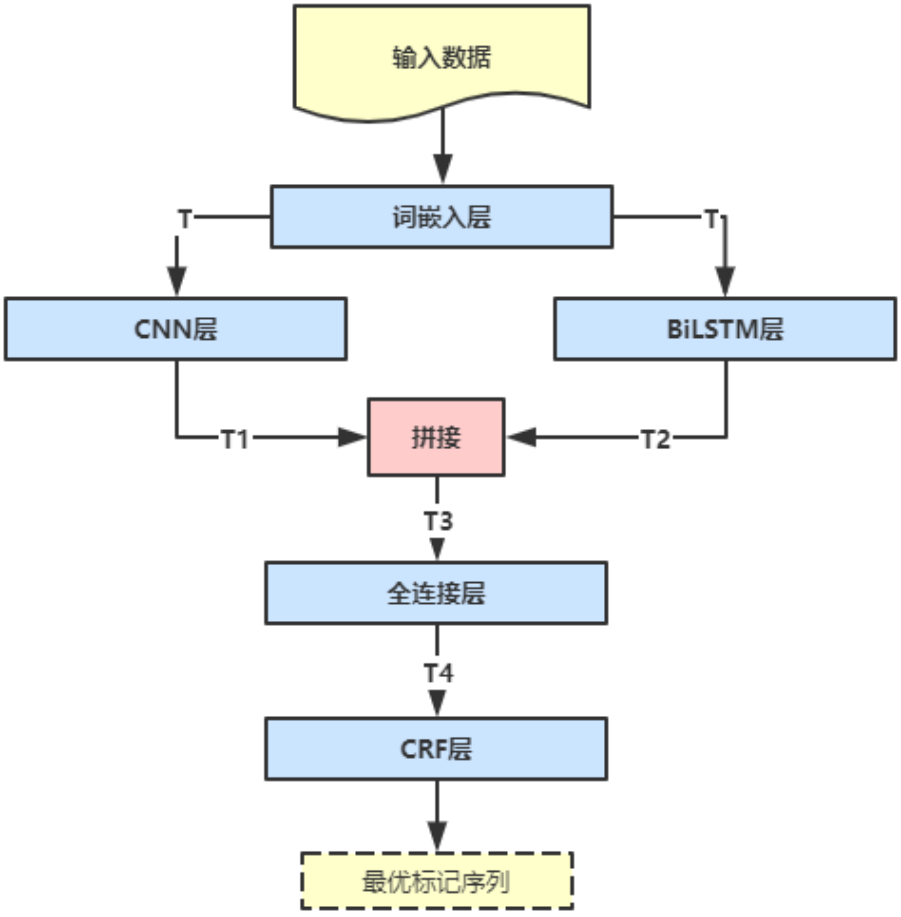

图 3.5 BiLSTM-CNN-CRF 模型框架

经过 Keras Embedding 完成文本向量化后，CNN 模型提取标记的局部信息 T1，BiLSTM 模型提取标记的全局信息 T2，将字符向量拼接为 T3 输入至全连接成得到 T4 后输入 CRF 层。全连接层将通过 BiLSTM 和 CNN 处理获得特征数据整合结果映射到样本空间中完成加权提纯操作。CRF 层综合转移矩阵和全连接层的标记向量结果，计算标记的得分并将最高分序列作为最终标记序列。本文使用的知识抽取算法 BiLSTM-CNN-CRF 详细流程如表 3.4 所示：

表 3.4 基于 BiLSTM-CNN-CRF 的知识抽取算法

算法：基于 BiLSTM-CNN-CRF 的知识抽取算法
输入：经人工标注的招聘需求文本 T、未标注实体的招聘需求文本集合 I 输出：完成实体标注的数据集 R 方法： （1）将标注集 T 使用 keras embedding 转换为中文字符级向量集 （2）字向量分别输入至 CNN 层和 BiLSTM 层提取标记的局部和全局特征 （3）模型初始化，配置并训练模型参数 （4）CNN 层输出 T1 词向量，BiLSTM 层输出 T2 词向量 （5）拼接 T1 和 T2 为 T3 字符级词向量作为全连接层的输入 （6）T3 输入全连接层提取文本特征并映射至样本空间中，完成特征的加权提纯输出 T4 （7）CRF 层综合转移矩阵和（6）的预测结果 T4 训练模型获得最优序列 （8）未标注实体的数据集 I 输入至（7）中输出完成知识标注的数据集 R

3.4.3 知识抽取实验结果与分析

本文实验引入召回率 REC（Recall）作为衡量识别结果与标注数据的占比，引入准确率 PRE（Precision）作为衡量正确结果与识别结果的占比，为了更加综合地评价实验效果，本文引入 REC 和 PRE 的加权调和平均值F_β值作为实验的全面评定指标，在其中加入 PRE 和 REC 的计算结果，其具体的计算公式如下：

$$F_\beta = (1+\beta^2)\frac{REC \times PRE}{\beta^2(REC+PRE)} \quad (3.1)$$

本文采取$\beta = 1$即计算F_1值，计算公式如下：

$$F_1 = \frac{2 \times PRE \times REC}{PRE+REC} \quad (3.2)$$

准确率PRE的公式定义为：

$$PRE = \frac{\text{正确识别的实体总数}}{\text{识别出的实体总数}} * 100\% \quad (3.3)$$

召回率REC的公式定义为：

$$REC = \frac{\text{正确识别的实体总数}}{\text{人工标注的实体总数}} * 100\% \quad (3.4)$$

模型的参数调节对训练结果起到一定的作用，其中包括学习率、滤波器数量和尺寸等等。其中具体数值的设置都将影响模型的训练结果，例如 CNN 层的学习率的大小影响更新速度和最优解的结果。是否使用随机失活（dropout）同样将影响模型效果，为了提升模型的泛化能力，本文在模型训练阶段使用 dropout 函数使每个神经单元以一定取值的概率被保留，避免在神经网络中出现过拟合现象。

在调试参数过程中，本文先对 dropout 分别取 0.1、0.3 和 0.5 作为对比参数，在 15 轮训练过程中统计不同取值在每轮中的 F1 值，正如图 3.6 所示在训练 6 轮过后三组 dropout 取值的 F1 值趋于稳定且效果差距较小，所以本文将 dropout 取值为 0.1。

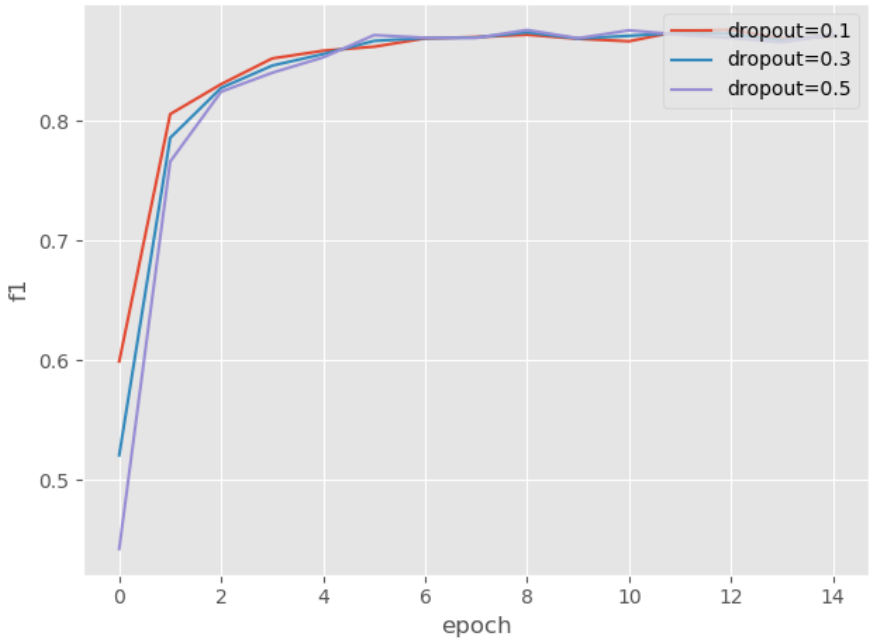

图 3.6 dropout 参数调节分析图

滤波器的数量和尺寸对 CNN 模型的效果存在影响，滤波器数目过多数据样本将出现过拟合，数目过少则会欠拟合；滤波器尺寸过大将无法过滤冗余信息，过小则会造成数据残缺。在保证其他参数不变的情况下，本文将滤波器的个数分别设置 10 个、20 个，对比实验显示当滤波器为 10 个和 20 个的 F1 值没有明显差别，所以将滤波器个数缩小至 10 以内，记录多轮训练过程中不同滤波器的 f1、准确率 pre 和召回率 rec，通过计算以上评价标准的最大值 max 和平均值 ave 生成图 3.7 中的分析结果，结果显示当滤波器的个数为 6 时取得较好的效果，各项评价标准均取得最大值。

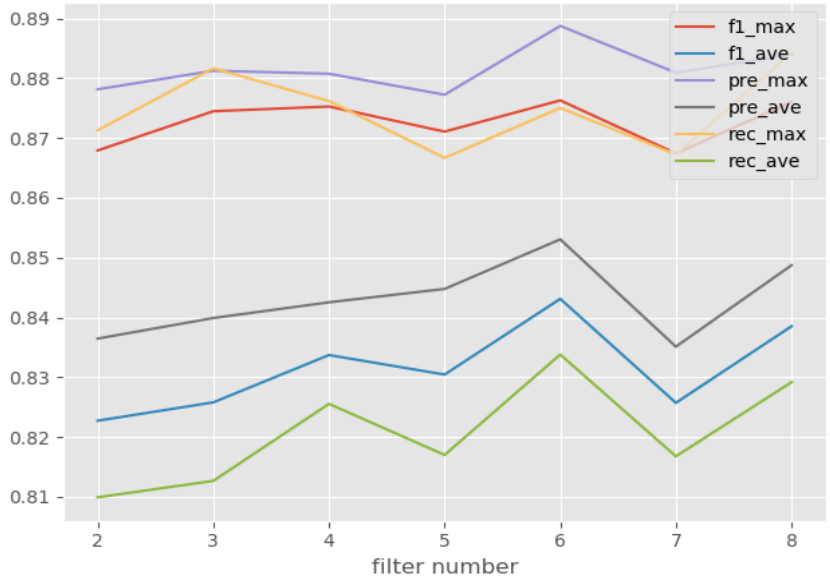

图 3.7 滤波器参数调节分析图

对 CNN 模型中卷积核参数调节过程中，将 dropout 值固定为 0.1，滤波器个数固定为 6，将卷积核数值范围限制在 2 至 10 之间统计每轮训练的准确率、召回率和 F1 值并记录最大值 max 和平均值 avg，最终实验结果如图 3.8 所示，经分析卷积核设置为 5 时训练效果达到最佳。

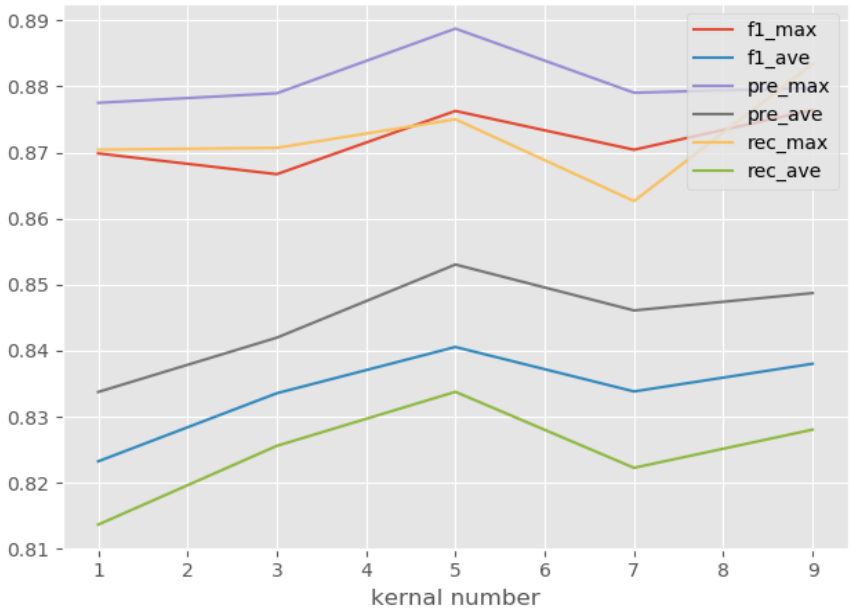

图 3.8 卷积核调节参数分析图

通过调参并对比实验结果确定了模型中的主要参数，对模型训练 15 轮并记录知识抽取模型的实验结果，其中包括 F1、PRE 和 REC 的每轮数值，由图 3.9 可见训练结束后模型的 PRE、REC 和 F1 值稳定在 85%以上。

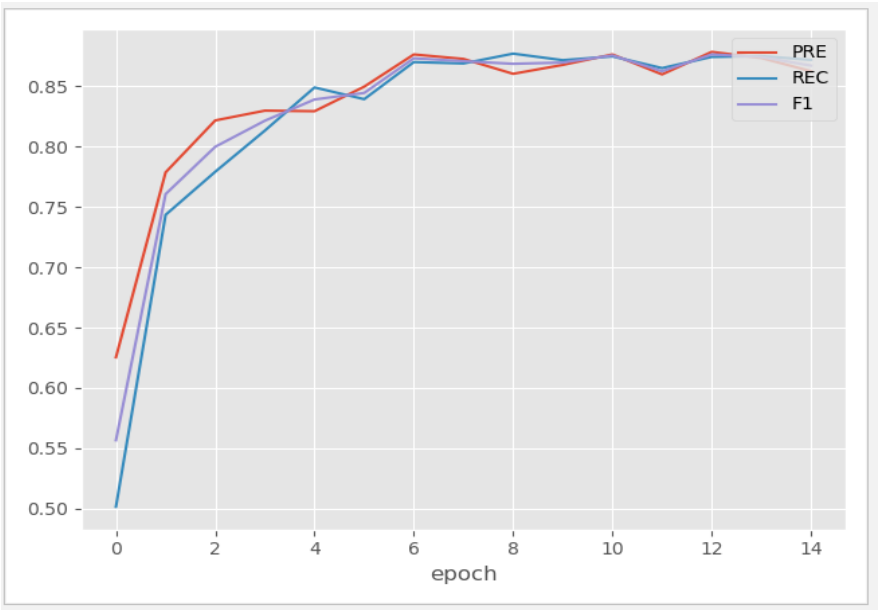

图 3.9 BiLSTM-CNN-CRF 模型训练结果图

经过 BiLSTM-CNN-CRF 模型识别 6 种实体后在每个实体后标记实体的属性，其中包括以 d-掌握程度、a-算法、t-工具、f-技术领域、p-编程语言和 o-其他内容的 6 种实体。本节实验共计输入 2000 条招聘需求描述文本，除去 o-其他内容类型实体后总计识别 62358 个实体，平均每条招聘需求信息中抽取 31.179 个实体，其中每类实体的具体数目和平均数目如图 3.10 所示。

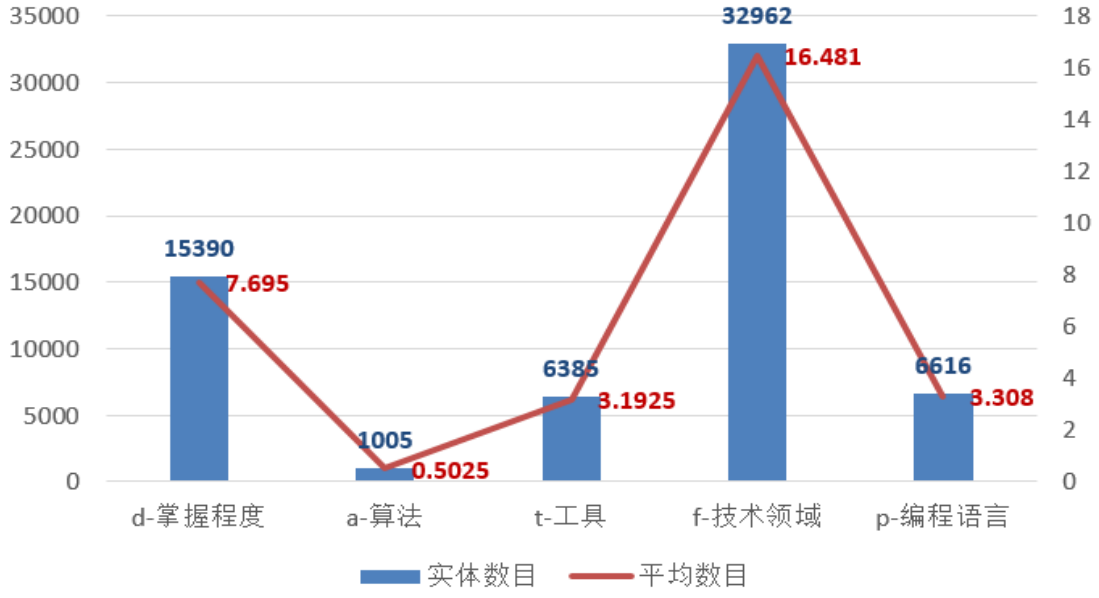

图 3.10 实体识别结果图

3.4.4 基于 Neo4j 的知识存储

完成知识抽取后，本小节采用 Neo4j 图数据库存储计算机领域招聘信息中的知识，在图数据库中创建节点、属性和关系。本文将 csv 格式数据导入 Neo4j 实现数据的导入操作，使用 Cypher 创建具体的节点、关系。在表 3.5 中展示使用 Cypher 创建节点和关系的具体语句。

表 3.5 Cypher 创建节点和关系语句

Cypher 操作	语句
创建节点	LOAD CSV WITH HEADERS FROM "file:///field.csv" AS line MERGE (f:field{id:line.id,field:line.field})
创建“岗位招聘-属于-领域”关系	LOAD CSV WITH HEADERS FROM "file:///re.csv" AS line match (from:work{id:line.work_id}),(to:field{id:line.field_id}) merge (from)-[r:属于{work:line.work,field:line.field}]->(to)

创建成功后通过“MATCH p=()-[r:`属于`]->() RETURN p LIMIT 100”查询关系后结果如图 3.11 所示，其中蓝色节点代表招聘需求，紫色节点代表在描述需求中提取的技术领域实体，连线代表“属于”的关系，即招聘需求属于技术领域，一个技术领域可以与多个招聘需求相连。

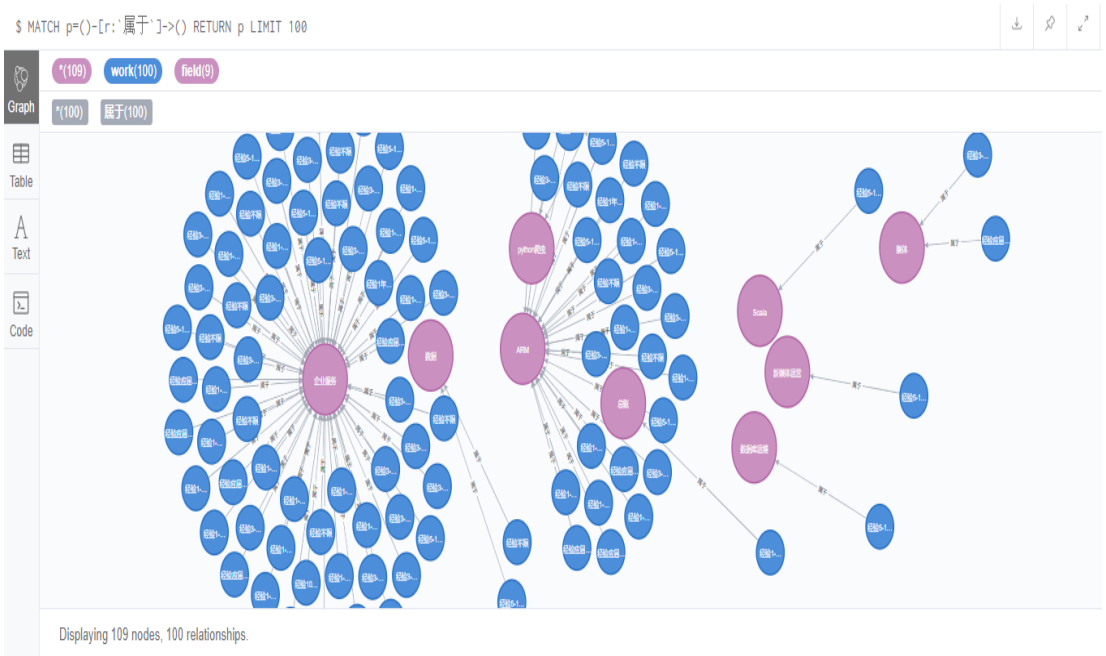

图 3.11 查询“属于”关系

本文将文本数据中的实体和关系存入 csv 文件再导入 Neo4j 中，经过预处理后的数据中包含 6 类实体和多种关系，总计节点 67422 个实体，27037 条关系，其中不同颜色代表不同的实体类型，不同的颜色的连线代表不同的关系类别。由于图谱中实体和关系较为丰富，图 3.12 仅展示 Neo4j 中部分实体和关系构成的子图，其中每一个实体节点可进一步扩展展示与其相关的实体和关系。

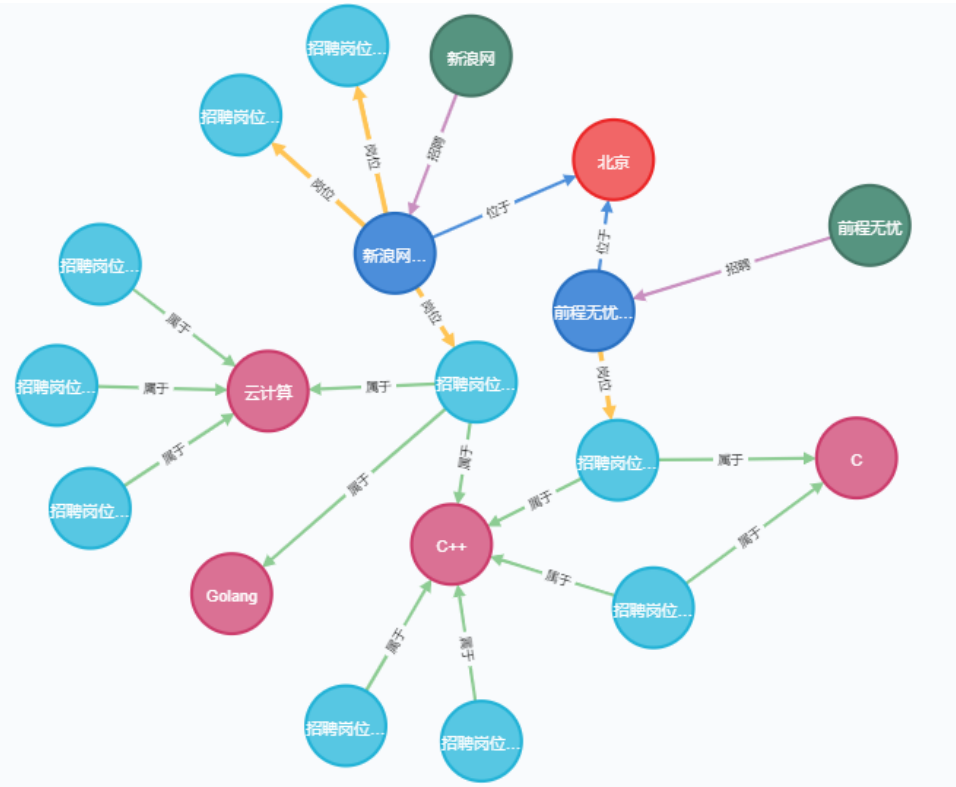

图 3.12 查询 Neo4j 中部分子图

3.5 本章小节

本章首先介绍基于计算机领域招聘信息的知识图谱的构建框架，从总体架构层面阐述构建图谱所用的方法和技术。然后分步骤构建图谱，设计数据模式小节中结合计算机领域招聘需求的特点，设计实体和实体间的关系，作为知识图谱的数据模式。编写 Python 爬虫获取计算机领域的招聘信息，借助人工标注方式标注部分实体和关系完成数据预处理任务。接着介绍基于 BiLSTM-CNN-CRF 的知识抽取模型和实验结果，而后将知识图谱中的实体和关系存储至 Neo4j 图数据库并实现图数据的相关操作。

第 4 章　岗位胜任力需求模型的构建

本章将基于前文的工作构建岗位胜任力需求模型，一是将知识抽取后的招聘需求文本通过 BERT 文本分类模型实现计算机领域岗位的分类，二是通过 word2vec 对招聘需求文本中描述胜任力的程度词分类，三是通过词共现矩阵抽取不同岗位所需的胜任力，主要包括知识和技能及其掌握程度，最终构建计算机领域不同岗位的胜任力需求模型。

4.1 胜任力需求模型的构建框架

参照岗位胜任力的定义将不同岗位所需的知识和技能作为胜任力要素，结合计算机领域求职网站中招聘信息的数据特征，依托知识图谱抽取计算机领域不同岗位所需的胜任力，其中包含计算机领域岗位所需的知识和技能及其对应的掌握程度，例如前文构建知识图谱中的掌握程度、算法、工具、编程语言等实体。在胜任力需求模型中引入不同类别程度词，以描述对不同岗位所需知识和技能的需求程度。胜任力需求模型构建框架如图 4.1 所示，最终抽取出计算机领域中不同类别岗位所需的知识和技能及对应的需求程度，帮助求职者更全面了解岗位。

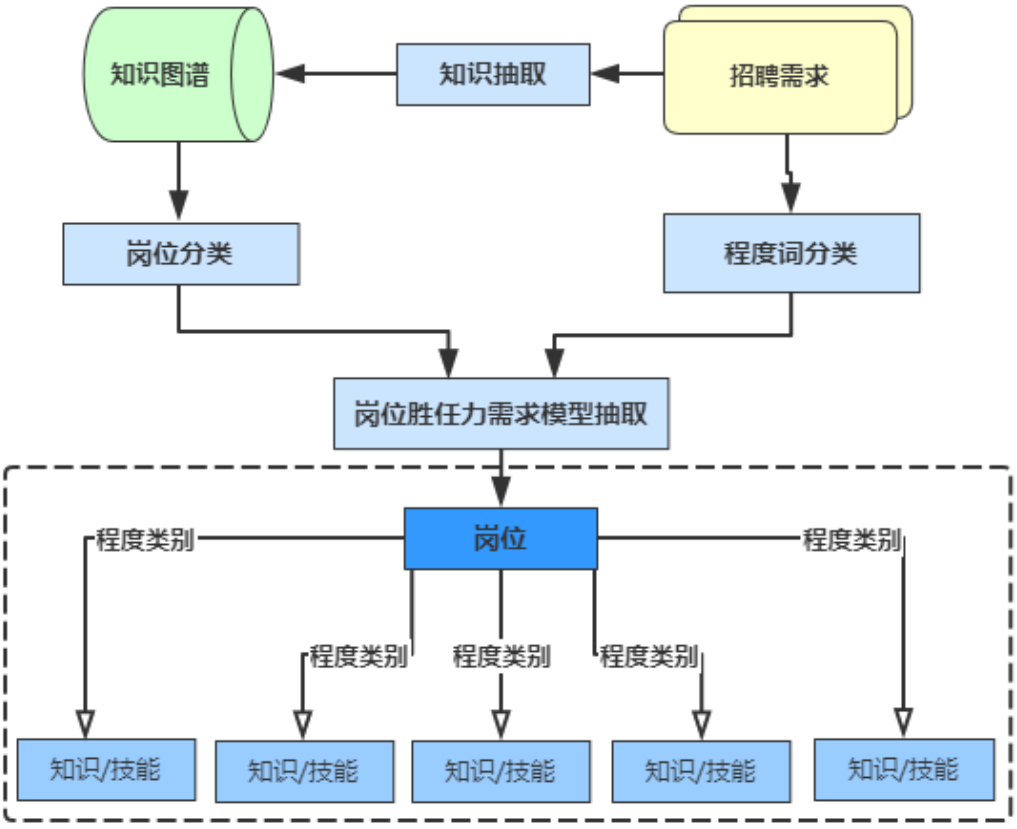

图 4.1 计算机领域岗位胜任力需求模型的构建框架

4.2 计算机领域岗位分类

4.2.1 基于知识图谱的文本语义扩展

本文采集的计算机领域招聘信息为多源异构数据，尤其在招聘需求描述这类短文本中具有内容短小、噪声大、主题不明等特点，其中存在包含公司简介和其他与招聘需求无关的数据，这样的数据特征稀疏影响岗位分类效果。本小节基于前文的知识抽取方法抽取出招聘需求描述文本中的中文实体和英文实体，但是由于英文实体中存在词形变化，所以本文使用基于概念图谱的实例概念化方法，获得英文实体的同概念词作为扩展实体加入至原始短文中以避免英文实体由于词形的变化影响岗位的分类效果，具体算法如表 4.1 所示。

表 4.1 基于知识图谱的语义扩展算法

算法：基于知识图谱的语义扩展算法

输入：招聘信息数据集 D、补充词的数目 topK，实例概念化算法 A
输出：语义扩展后的短文本D^*
方法：
（1）$\text{Words} = KG(D_i)$;//对原始文本数据集中的每一条短文抽取出知识图谱中实体构成初始特征集Words
（2）$\text{I} \leftarrow \text{selectEnglishNode(Words)}$;//抽取特征集 Words 中英文实体，组成概念图谱中实例集合 I
（3）$I_i \leftarrow Words_i \cap \text{I}$;//抽取出每条短文本中需要概念扩展的实体
（4）$EWord_i \leftarrow AccessAPI(I_i, A, topK)$;// $I_i, A, topK$拼接为 url，调用微软概念图谱 API 获取 topK 个扩展实体
（5）$Words_i\_new \leftarrow Combine(Words_i, EWord_i)$;//将扩展后的实体拼接
（6）$D^* \leftarrow Data(Words_i\_new)$;//将每一条实体拼接后的短文整合为扩展后的短文集合D^*

其中引入的实例概念方法 A 为用标准化的点互信息（Normalized Pointwise Mutual Information，NPMI）和概念推理算法（Basic-level Categorization，BLC）计算实体评分以及概念的评分[53]，最终返 topK 个概念词，具体公式如下：

$$NPMI(i,c)=\frac{PMI(i,c)}{-\log P(i,c)}=\frac{\log P(i|c)-\log P(i)}{-\log P(i,c)} \quad (4.1)$$

$$BLC(i)=arg\max_{c}\max_{c} Rep(i,c) \quad (4.2)$$

$$Rep(i,c)=P(c|i)\cdot P(i|c) \quad (4.3)$$

其中，i 代表在文本中的实例，c 代表在概念图谱中的概念，$P(c|i)$为文本中的 i 所对应 c 的概率，$P(i|c)$为 c 所对应的文本中 i 的概率。

表 4.2 语义扩展示例表

知识抽取($Words_i$)	d:负责 f:视频图像 f:算法 d:负责 f:图像分类 f:视频内容理解 f:人脸表情 f:视觉算法 d:参与 f:视觉算法 d:协调 f:视觉算法 d:丰富 f:数据方向 t:toC d:熟悉 f:深度学习 f:开发框架 t:Tensorflow t: PyTorch t:Caffe t:Theano t:MXNet d:熟悉 f:业务
扩展结果($Words_i\_new$)	负责，视频图像，算法，负责，图像分类，视频内容理解，人脸表情，视觉算法，参与，视觉算法，协调，视觉算法，丰富，数据方向，toC，熟悉，深度学习，开发框架，Tensorflow，PyTorch，Caffe，Theano，MXNet，熟悉，业务，application ，analysis tool，analysis，popular deep learning library， great dnn toolkit，deep learning implementation framework，open source program，toolkit，source package，neural network implementation
英文实体“Theano”扩展结果I_i	{ "deep learning framework": 0.14705882352941177, "framework": 0.11764705882352941, "open source eorts": 0.11764705882352941, "library": 0.058823529411764705, "open source tool": 0.029411764705882353, "popular deep learning library": 0.029411764705882353, "source package": 0.029411764705882353, "python module": 0.029411764705882353, "popular deep learning software": 0.029411764705882353 }

4.2.2 基于 BERT 模型的岗位分类

Google 提供了针对不同语言的预训练 BERT 模型，本文引入 [Bert-Base-Chinese]字符级的中文模型，其中包括训练使用的中文词典。模型训练阶段分为加载预训练模型、训练模型、循环迭代利用验证集调整模型参数和保存最佳模型参数。完成训练阶段后进出入测试阶段，具体流程如图 4.2 所示。

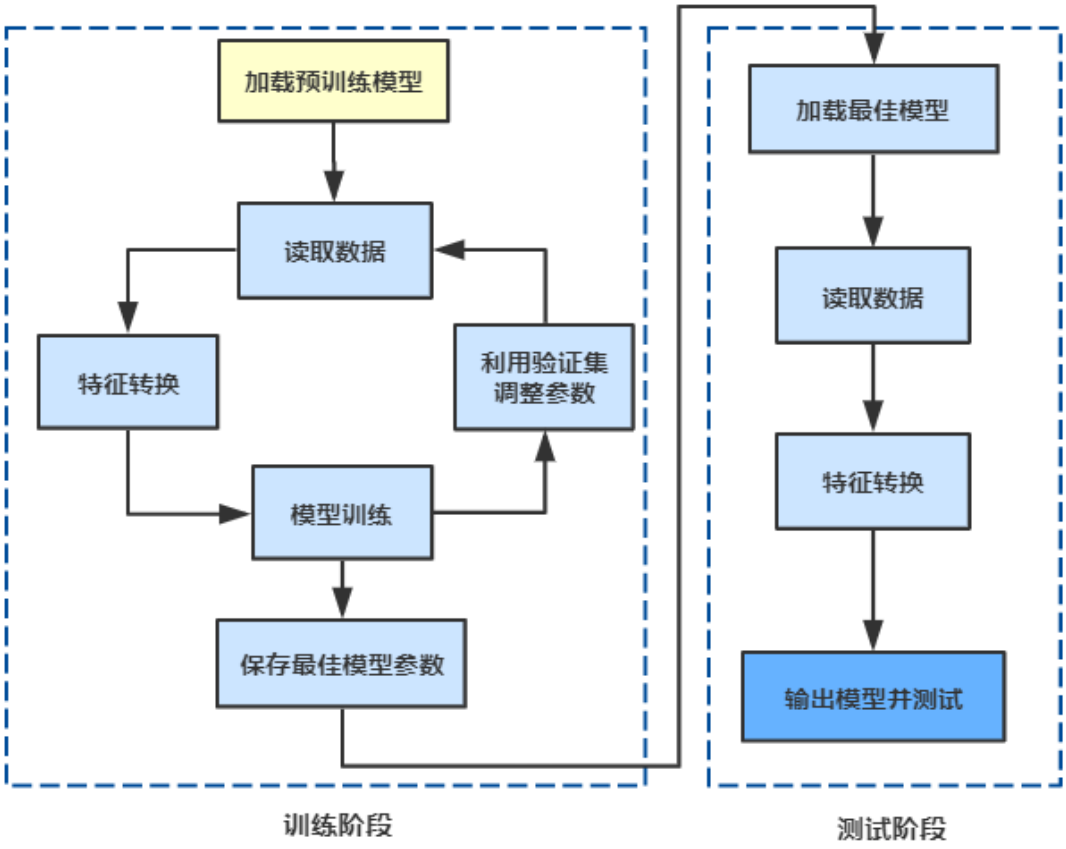

图 4.2 BERT 模型关键流程图

本文将 600 条计算机领域招聘信息中的需求描述文本合并作为数据集，参照互联网中计算机领域招聘网站对于岗位的分类和计算机领域专家的意见，通过人工标注的方式将数据集分类为人工智能类、自然语言处理类、深度学习类、软件工程类、数据库类和操作系统类。将已标注的数据按照 6:2:2 的比例分为训练集、验证集以及测试集。数据完成知识抽取和语义扩展后存储为 tsv 数据格式，包含“类别”和“文本”两列数据，作为 BERT 模型数据输入文件。

BERT 模型的数据读取模块包含数据基类和基于数据基类定义的数据读取类，在数据读取类中针对不同的数据集完成读入操作，最终返回包含序号、中文文本和类别的列表作为特征转换的输入文件。读入数据后在文本前后加入[CLS]和

[SEP]标记文本的开始和结束，分词后将字转为字典中对应的 input_id，用“input_mask”标记真实字符/补全字符，真实字符为 1，补全字符为 0。表 4.3 为使用 BERT 模型处理后的部分文本样例。

表 4.3 本文的 BERT 模型文本表示样例

tokens:	[CLS] 负 责 自 然 语 言 理 解 研 发 工 作、文 本 数 据 的 挖 掘 分 析, 参 与 对 话 机 器 人 … [SEP]
input_id:	101 6566 6569 5632 4197 6427 6241 4415 6237 8020 156 10079 6569 3152 3315 3144 2945 4638 2905 2963 …
input_mask:	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 ……
label:	自然语言处理 (id = 1)

文本转换为特征值后将用于模型的训练和测试，模型的训练中的模型参数是在多次训练中不断优化的，训练算法为 BERT 专用的 Adam 算法，基于训练数据迭代更新神经网络的权重，解决系数梯度和噪声问题。在每一轮训练后会在验证集中测试并得出相应的 F1 值，如果 F1 值大于此前最高分则保存模型参数，训练截止后将保存最佳模型参数。本文在训练 BERT 模式时对主要参数进行初始化，其中 L2 权重衰减为 0.01，在预热训练中线性增加学习率，剩余 BERT 模型的部分参数如表 4.4 所示。

表 4.4 BERT 模型初始化参数列表

参数名	注释	类型	值
bert_model	预训练模型	string	bert-base-Chinese
train_batch_size	训练时 batch 大小	int	64
eval_batch_size	验证时 batch 大小	int	1
learning_rate	Adam 初始学习步长	float	5e-5
seed	初始化时的随机数种子	int	777

本小节的实验环境为 Python3.6，基于 PyTorch 深度学习框架和 Keras 的 tokenizer 分词读入文本数据，使用 Masked LM 构建语言模型，随机遮盖或替换文本中的任意字，训练模型通过上下文预测被遮盖或替换的部分。文本中 15%中文字符（token）被随机替换，其中 80%替换为[mask]、10%替换为其他 token、

10%不做改变。经过训练阶段，模型最终输出的隐藏层的计算结果的维度为X_{hidden}:$[batch_size, set_len, embedding_dim]$，初始化一个映射层的权重$W_{vocab}$完成隐藏维度到字向量数量的映射，最后求得$X_{hidden}$和$W_{vocab}$的矩阵乘$X_{hidden}W_{vocab}$:$[batch_size, seq_len, vocab_size]$后将结果在 vocab_size 维度上做 softmax 归一化，通过 vocab_size 里最大概率值来得到模型的预测结果。其中 batch_size 设置为 64，seq_len 设置为 202，vocab_size 为 21128。

本文 Bert 模型的训练共计 10 轮，使用准确率和召回率的加权调和平均值 F1 值作为实验效果的衡量标准，在模型训练过程中统计每次 F1 值，如图 4.3 所示。经过数据汇总，本小节模型在第 9 次训练结束后 F1 值趋于稳定，在第 6 次的 F1 值较高，根据 Adam 算法本次的训练参数将被选取为 BERT 模型最终参数保存并应用在后续的计算机领域岗位分类中。

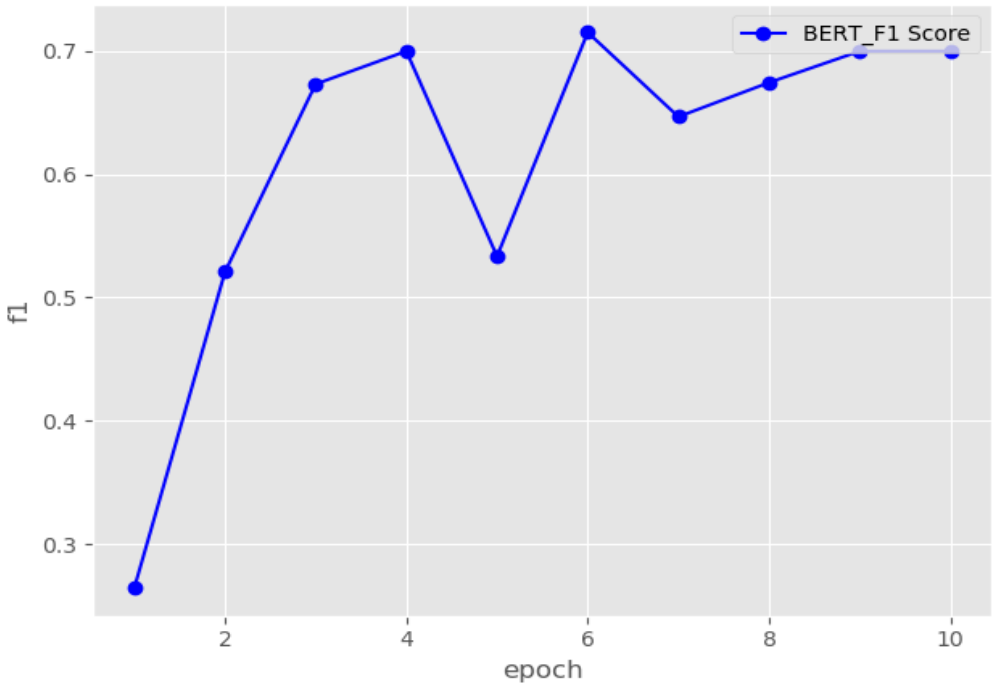

图 4.3 BERT 模型训练阶段 F1 值统计图

4.2.3 实验结果与分析

完成模型训练后向模型输入六类岗位的描述文本进行测试，在已完成人工类别标注的岗位招聘需求描述数据按照类别各抽取 20 条共计 120 条数据作为测试集。对岗位划分结果及各类划分数目（support）进行输出，得出表 4.5 中结果。除此之外，为了全面综合地衡量 BERT 模型的分类效果，本文通过 sklearn 库中

的 classification_report 函数计算出针对实验 F1 值的宏平均数值，经计算实验最终的 macro_avg（f1 score）为 0.63。

表 4.5 计算机领域岗位测试集分类结果

类别	precision	recall	f1 score	support
深度学习	0.60	0.80	0.69	15
自然语言处理	0.70	0.78	0.74	18
人工智能	0.65	0.54	0.59	24
软件工程	0.85	0.50	0.63	18
数据库	0.75	0.52	0.61	29
操作系统	0.45	0.56	0.50	16

从实验结果可见操作系统的分类结果较低，原因在于操作系统岗位的部分招聘需求文本中出现较多的与自然语言处理和人工智能岗位相关的词语，导致 BERT 模型无法准确区分操作系统岗位的招聘信息。本小节最后将未标注的 2098 条招聘需求完成岗位分类任务，其中岗位分类结果如图 4.4 所示。

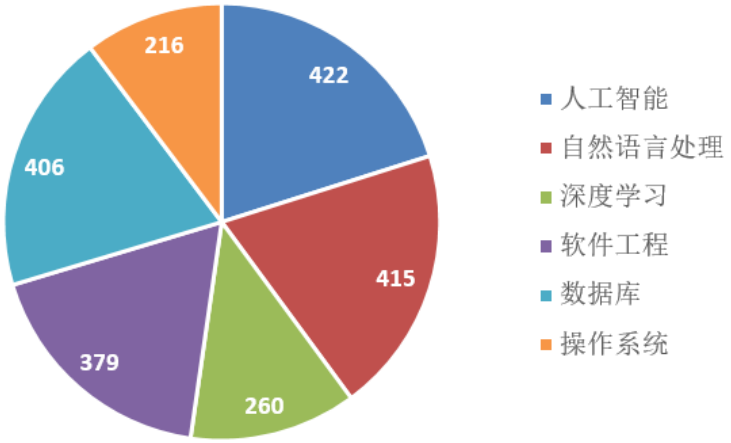

图 4.4 计算机领域岗位分类占比图

对实验进行分析可以看出 BERT 模型在文本分类中取得效果和数据质量相关，此外 BERT 模型在对中文文本分类是基于对字进行拆分，这种数据处理模式削弱了词语间的关联从而降低了岗位分类准确率。除了 BERT 模型的处理数据机制问题，数据倾斜问题也是本文岗位分类任务的挑战，例如在操作系统中出现很多关于自然语言处理的文本，导致个别类别的分类效果较差影响分类器效果，未来将针对数据倾斜的问题进行改进。

4.3 程度词分类与胜任力需求模型抽取

4.3.1 基于 word2vec 的程度词分类

在招聘需求描述中的程度词包含很多种，例如“负责”、“熟悉”、“精通”、“具备”、“具有”、“了解”、“熟练”等等。前文的知识抽取可抽取出来程度词实体，但是有很多程度词表示的语义比较相近，例如“具有”和“具备”表示的语义基本一致，如果将所有程度词都展示在岗位所需的胜任力中将出现冗杂的数据展示效果，所以本文采用计算词语相似度的方法对程度词分类。首先抽取出计算机领域中招聘需求描述里所有程度词，基于 word2vec 对程度词进行词语余弦相似度计算，根据排序结果将程度词分类，具体算法如表 4.6 所示。

表 4.6 基于 word2vec 的程度词分类算法

算法：基于 word2vec 的程度词分类算法

输入：数据集 D、界定词 BorderWord、程度词分类数目 ExtentN
输出：程度词相似度排序集合 EList
方法：
（1） $\text{WordList} = KG(D)$;//对招聘需求文本完成实体抽取，得到初始程度词语集合 WordList
（2） $\text{Word}_{\text{vec}} \leftarrow$Word2vec（WordList）;//对 WordList 根据 word2vec 优化后的 CBOW 模型获得词语向量化表示$Word_{vec}$
（3） $\text{ScoreList} \leftarrow \text{SimilarScore}(\text{BorderWord}, \text{Word}_{\text{vec}})$;//统计每一个$Word_{vec}$与$BorderWord$的相似度
（4） $\text{List} \leftarrow \text{Sort}(\text{ScoreList})$;//将计算相似度结果$ScoreList$降序得到程度词语$List$
（5） $\text{Elist} \leftarrow \text{splitExtent}(\text{List}, \text{ExtentN})$;// 根据相似排序结果 List 将程度词List分为 ExtentN 类程度词

本文使用基于 word2vec 模型的程度词分类算法对程度词实体进行相似度的计算和分类，程度词“精通”作为界定词取值为 1，通过模型最后得出其他程度词与“精通”的语义相似度，部分程度词语义相似度结果展示如表 4.7 所示。

表 4.7 部分程度词排序列表

程度词	与“精通”相似度
精通	1.00000000
熟练掌握	0.66228044
善于	0.64182961
运用	0.614020705
理解	0.605181098
掌握	0.588120699
从事	0.582218945
熟练	0.560444713
具有	0.557942212
具备	0.547494173
使用	0.530996442
研究	0.496722817

实验最终抽取出 60 个程度词，与“精通”相似度值在[0.05,1]区间内，排序后在每类程度词中选取相似度中位数的词作为代表词命名程度词类名，程度词被分为掌握、熟悉、负责、了解和参与五类，具体如表 4.8 所示。

表 4.8 程度词分类结果

类别	与“精通”相似度	程度词
掌握	(0.49,1]	精通、熟练掌握、善于、运用、理解、掌握、从事、熟练、具有、具备、使用、研究
熟悉	(0.39,0.49]	拥有、完善、指导、应用、富有、熟悉、设计、结合、根据、改进、综合、分析
负责	(0.30,0.39]	参与、利用、提供、通过、基于、负责、提升、开发、识别、独立、主持、主导
了解	(0.19,0.30]	追踪、建立、支持、进行、尝试、了解、参加、完成、探索、优化、构建、深度
参与	(0,0.19]	挑战、梳理、调研、思考、推动、参与、搭建、至少、跟进、带领、挖出、持续

4.3.2 基于共现矩阵的胜任力需求模型抽取

在招聘需求文本中程度词与胜任力一同出现的次数较多则表示该类岗位需求相应程度的胜任力，例如在自然语言处理这类岗位中频繁出现“掌握 Python”则表示自然处理这类岗位对 Python 这种技能的需求程度为“掌握”。本文计算机领域岗位胜任力需求模型的构建方法是统计在各类岗位招聘需求描述中程度词与胜任力实体同时出现的次数来抽取岗位所需的不同程度的胜任力。技术方面

则通过生成词共现矩阵来完成抽取文本中知识和技能实体与程度词的共现关系，共现矩阵结构如图 4.5 所示。本文对共现矩阵计算方式进行了改进，在共现矩阵计算过程中针对不同掌握程度类别赋予不同的语义权重，并将词频与语义权重的乘积结果加入至共现矩阵的最终结果。生成的共现矩阵可以较为全面地展示不同岗位对胜任力的需求程度，但是为了更加简洁表示胜任力需求模型，本文将各个胜任力实体与不同类别掌握程度实体矩阵结果求和，获得该类岗位对于不同胜任力的综合需求程度。

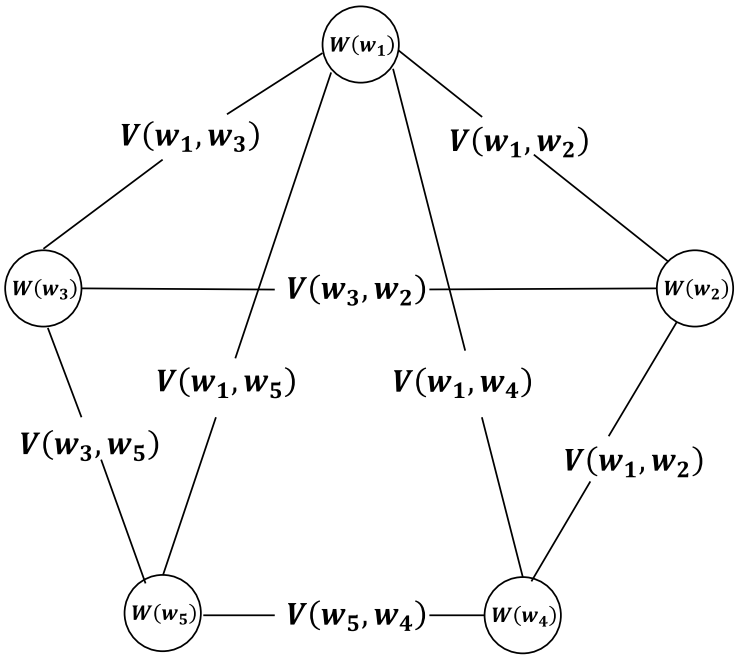

图 4.5 共现矩阵结构图

词共现矩阵如果简单统计频率可能会抽取出与岗位胜任力无关的词汇，所以本文对程度词出现的频率和语义综合考虑，重新定义程度词权重。因为掌握类、熟悉类、负责类、了解类和参与类程度词具有不同的掌握程度的语义，所以本文在构建词共现矩阵时重新定义程度词的权重，最终的权重公式如下所示。

$$FinalWeight(w_i) = Weight(w_i) + \sum_{j=1,j\neq i}^{n} v(w_i, w_j) \quad (4.4)$$

$$Weight(w_i) = tf(w_i) \times f_{sw}(w_i) \quad (4.5)$$

式中：$tf(w_i)$代表程度词w_i的词频，$f_{sw}(w_i)$代表w_i的语义权重，其中掌握类程度词：1.0，熟悉类程度词：0.8；负责类程度词：0.6,；了解类程度词：0.4；参与类程度词：0.2。

通过为计算机领域六类岗位的招聘需求文本集合构建程度词和胜任力实体的共现连通图，抽取程度词与胜任力实体间的关联关系，即获得每类岗位对于胜任力的需求程度，具体抽取算法如表 4.9 所示。

表 4.9 基于共现矩阵的胜任力抽取算法

算法：基于共现矩阵的胜任力抽取算法

输入：岗位的招聘需求文本数据集 D、程度词集合 E、程度词权重 W、程度词类别数据 N
输出：岗位所需胜任力M^*
方法：
（1） $Entities = KG(D)$;//对数据集 D 中的每一条文本数据进行知识抽取 KG(D)获得实体集 Entities
（2） $Entities_{competencey} = Entities - E$;//除去程度词 E 保留胜任力实体$Entities_{competencey}$，其中包含知识和技能等实体
（3） $M \leftarrow CoMatrice(Entities_{competencey}, E, W)$;//引入程度词权重 W 计算胜任力实体和程度词的共现矩阵 M
（4） $List_{competencey} \leftarrow Sum(M, Entities_{competencey})$;//计算 M 中各胜任力实体对不同类型程度词的权重和，得到胜任力列表$List_{competence}$
（5） $M^* \leftarrow Sort(List_{competencey}, N)$; //List 排序并 N 等分后输出$M^*$

4.3.3 实验结果与分析

本节实验基于前文的算法抽取出不同类别岗位中招聘需求所提及的胜任力集合与程度词的共现矩阵，并针对共现矩阵结果对各个胜任力实体综合计算权重，输出最终的岗位胜任力需求模型。接下来以自然语言处理类招聘信息为例，展示胜任力需求模型的抽取结果。本文抽取出自然语言处理岗位中 866 个胜任力实体，胜任力实体与掌握程度词间共现矩阵中的权重经求和后的取值区间为[0, 80.413]，其中部分共现矩阵结果如表 4.10 所示。根据前文定义的掌握程度类别对胜任力需求列表分成 5 类，部分岗位胜任力需求模型的结果如表 4.11 所示：

表 4.10 自然语言处理岗位胜任力与程度词部分共现矩阵

	掌握	熟悉	负责	了解	参与
自然语言处理	21	16.527	43.677	0.878	2.63
深度学习	12	6.296	37.98	1.317	2.893
NLP	18	9.444	34.815	0.878	2.367
机器学习	12	18.888	45.576	2.634	1.315
AI	0	0	0.633	0	0.789
Python	19	0	14.559	0	0
C++	34	3.148	10.761	0	0.789
Java	18	0.787	10.128	0	0
Pytorch	1	0	1.899	0	0
Torch	1	0	1.899	0	0
Tensorflow	5	0	10.761	0	0.263
LSTM	1	0	8.862	0	0.263
CNN	1	0	11.394	0	0.263
RNN	1	0	10.761	0	0
seq2seq	0	0	1.266	0	0
BERT神经网络模型	0	0	0.633	0	0
神经网络	1	0	0	0	0

表 4.11 自然语言处理岗位胜任力需求模型部分列表

序号	掌握	熟悉	负责	了解	参与
1	机器学习	实体识别	舆情分析	关系抽取	问答系统
2	NLP	聚类	句法分析	JAVA	算法设计
3	深度学习	信息检索	TensorFlow	Pytorch	推荐系统
4	C++	linux	分析问题	Torch	意图识别
5	知识图谱	词性标注	人工智能	文本分析	Keras
6	Python	文本挖掘	R	智能对话	智能问答
7	Java	shell	大数据	搜索引擎	序列标注
8	数据结构	机器翻译	LDA	caffe	云计算
9	C	分类	Shell	自然语言理解	大数据技术
10	数据挖掘	信息抽取	对话系统	知识抽取	异构数据库

本文引入基于词频统计的共现矩阵作为对比实验，此实验中只统计在各类岗位需求描述中程度词和胜任力实体共同出现的次数，不引入程度词语义权重。通过问卷调查的方式对不同方式构建的计算机领域岗位胜任力需求模型的结果进行评估。实验参与者为 20 名计算机学院具中有计算机领域求职或实习经历的硕士生，结合自身经验评价问卷中的岗位胜任力需求模型是否符合实际的求职情况。

以自然语言处理岗位胜任需求模型为例，具体问卷调查的结果如表 4.12 所示。

表 4.12 问卷调查统计结果

程度	非常符合	比较符合	一般符合	比较不符合	完全不符合
本文方法	10	5	2	2	1
词频方法	2	4	2	8	4

从具有求职或实习经历的计算机专业学生角度评估计算机领域岗位胜任力需求模型的结果图 4.6 中可见，本文方法的符合占比为 85%，基于词频的方法符合占比 40%，由此可见本文的计算机领域岗位胜任力模型较为符合求职的实际情况，较基于词频生成的岗位胜任力需求模型的符合实际情况提升了 45%。

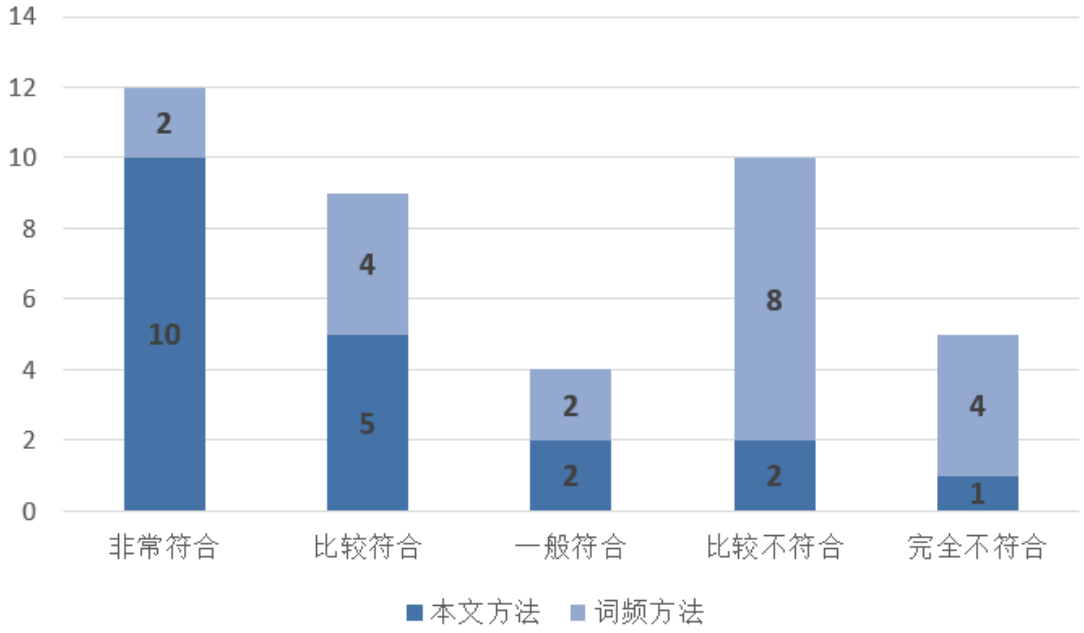

图 4.6 岗位胜任力需求模型的用户评价对比图

4.4 本章小结

本章首先提出了计算机领域岗位胜任力需求模型的抽取框架，接下来按照框架分为计算机领域岗位分类、程度词分类和胜任力模型抽取三个任务。基于知识图谱的短文扩展和 BERT 模型完成对计算机领域的岗位分类任务，基于 word2vec 完成程度词分类任务，通过共现矩阵完成胜任力需求模型的抽取任务。实验结果表明本章抽取的胜任需求模型较为符合求职的实际情况，可以为求职者了解计算机领域中的岗位提供借鉴和参考。

第 5 章　计算机领域胜任力管理平台的设计与应用

本章构建了一个基于计算机领域知识图谱的胜任力（Computer Science Competence Management）管理平台（以下简称 CSCM 平台）。CSCM 平台中关于招聘信息的知识和关系存储在 Neo4j 中，将知识图谱与岗位胜任力需求模型数据整合后，同步至 Elasticsearch 实现招聘信息的全文语义检索。基于 Vue.js 框架搭建前台功能和岗位胜任力需求模型展示模块。平台中的计算机领域知识图谱连接岗位、公司、胜任力、招聘需求等数据信息，为用户提供具有语义相关的检索结果，根据检索内容推荐相关的招聘信息，通过知识图谱可视化展示岗位所需的胜任力，实现数据列表和知识图谱的不同展示方式。

5.1 平台总体架构

CSCM 平台的总体架构分为知识图谱，后端搜索引擎和前端展示三部分。知识图谱由原来所在的 Neo4j 图数据库同步至 Elasticsearch（ES）便于全文检索，ES 主要分为索引层、Transport 层和 Restful API 三部分，各部分协同合作完成通讯与交互。平台前端基于 Vue.js 将响应数据和视图组件绑定，实现响应式数据处理平台。图 5.1 是 CSCM 平台的 B/S 架构图，后端通过 json 数据实现数据传输，前端通过 axios 向后台发送数据请求实现浏览器和 ES 的交互。

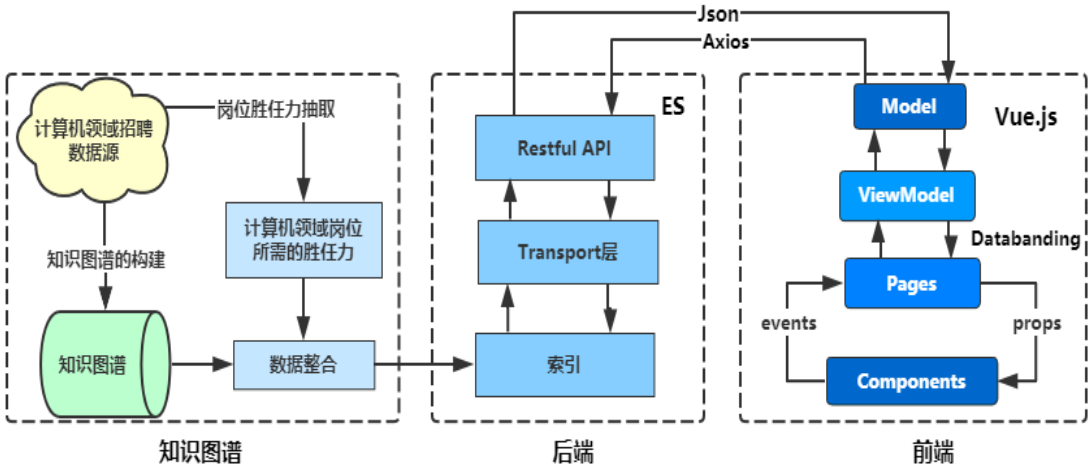

图 5.1 CSCM 平台架构图

经过知识抽取的数据存入 Neo4j 形成具有实体和关系的知识图谱，图谱数据同步至 ES 中便于与前台通过 json 数据交互并借助 ES 的搜索引擎的优势实现效果较好的搜索功能。其中在 ES 的数据是以 Node 集群的方式存储，增设副本保证

数据高可用性，将数据节点和传输节点隔离，分别处理数据和连接的问题。在 ES 中包含的索引类似于关系数据库中的库，通过基于 HTTP 的 Restful API 接口实现对 ES 中索引的查询、插入、修改、删除等操作。

平台基于 Vue.js 完成前端开发，视图层调用数据驱动的 Web 界面库函数，通过接口完成数据绑定和灵活的组件刷新。其中首先获取在 ES 中的数据，引入组件文件，在需要引入的组件文件中加入组件标签，最后通过声明式渲染完成数据显示。在前端展示模块中包含知识图谱和岗位胜任力需求模型的可视化，通过调用 Echarts 图表库在浏览器中展示平台的图谱数据。

前后端交互过程包括 ES 返回 json 数据和前端发送数据请求，其中 Vue.js 通过 Axios 向后台提交数据请求，Axios 提供了多种对后台数据请求的方式。前端发送的 JavaScript 对象序列通过 Axios 转化为 json 格式发送至 ES，前端收到 ES 回传的 json 数据将被解析为 JavaScript 对象发送至 Vue.js 中的组件完成数据的渲染，前后端的交互数据在 Axios 的转换下实现数据的查询和展示。

5.2 功能模块

CSCM 平台的功能包括基于知识图谱的计算机领域招聘信息全文搜索和岗位胜任力需求模型的可视化。全文检索基于 Elasticsearch 搜索引擎，同步 Neo4j 图数据后与前端交互实现对招聘信息的全文检索；Vue.js 将知识图谱按照实体和关系分类实现可视化，包括对招聘信息关联图谱和胜任力需求模型图谱。

5.2.1 基于 Elasticsearch 的全文检索

Elasticsearch（ES）是全文检索的主流软件之一，其底层为开源库 Apache Lucene，将 Lucene 封装后通过 Restful API 实现检索和存储。ES 中数据的每个字段都可以被索引与搜索，可以同时操作多个索引，可以存储 PB 级别的结构化或非结构化数据。在 ES 中的数据按照索引和类型完成分类存储，记录与文档（Document）相对应，文档中包含记录的属性和内容。与关系型数据库类比，ES 中的索引对应 DB 中的库，ES 中的类型对应 DB 中的表，ES 中文档对应 DB 中的行。ES 中的检索方式主要分为 Query String（以下简称 QS）和 Query Domain Specified Language（以下简称 QDSL）。QS 方式将查询的参数附带在 HTTP 请求的 query string 中，而 QDSL 方式是通过一个 json 格式的 HTTP 请求体作为条

件，请求体中包含复杂的查询组合。通过对存储在 ES 中索引为“eb_job”类别为“work”的岗位数据使用 QS 方式完成查询，图 5.2 显示通过 QS 查询方式获得的部分招聘信息数据。

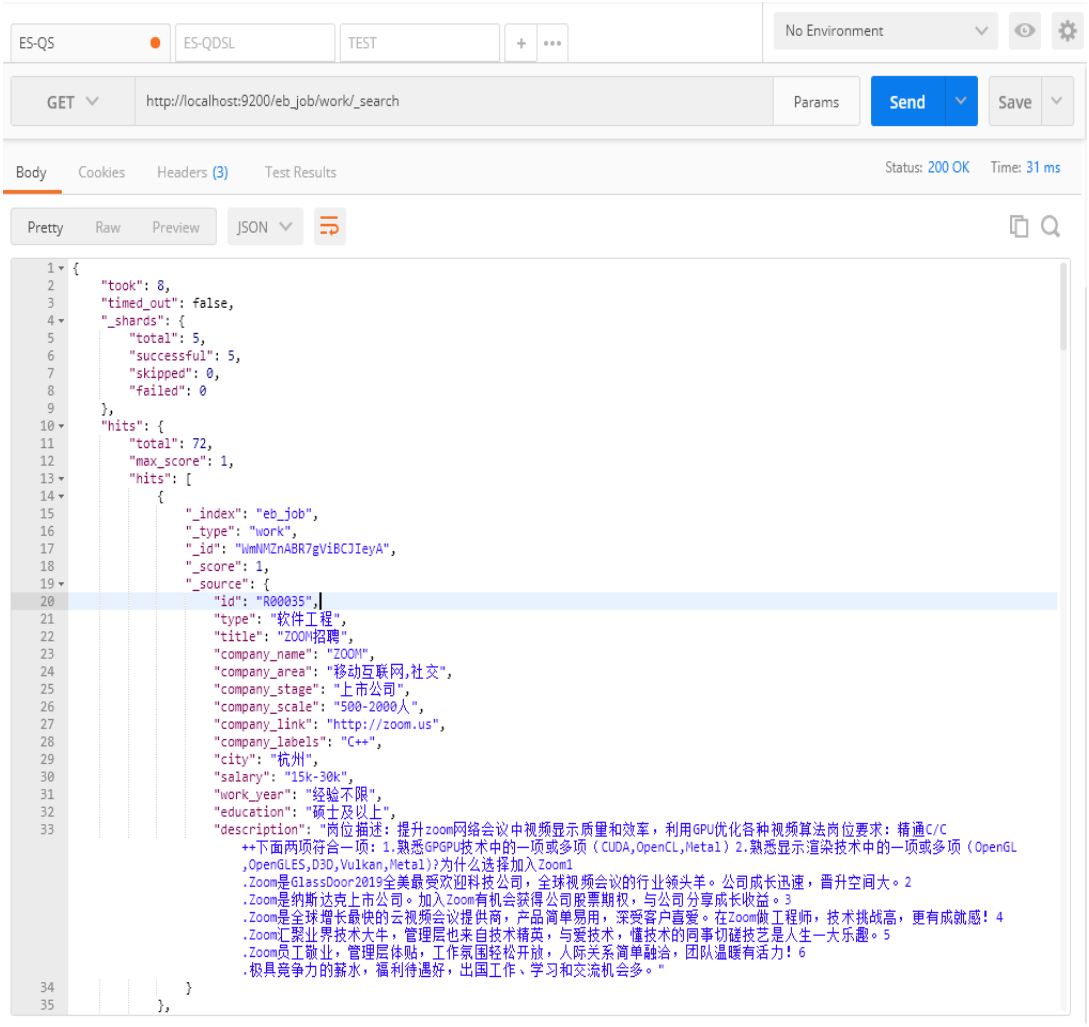

图 5.2 Elasticsearch 使用 QS 方式查询示例图

ES 返回的 json 数据已经按照与搜索词的相关度评分（_score）将返回的文档排序，文档的相关度评分取决于每个查询词在文档中的权重，词的权重由词频、逆向文档频率和文档长度归一值决定，标准计算法为 TF-IDF。

5.2.2 基于 Vue.js 的数据展示

本文搭建的 CSCM 平台基于 B/S 架构，其中前端使用 Vue.js 框架，通过 Axios 实现与后端的交互。Vue.js 框架将每个网页中所展示的数据按照类别形成组件，在加载页面时引入组件，减少代码冗余，父组件通过属性 props 向下传递数据给子组件，子组件使用事件 events 给父组件发送消息，通过 DataBinding 自动刷

新控件同步数据。在数据展示功能中包括搜索结果列表展示和知识图谱可视化，其中搜索结果通过列表的形式展示岗位信息及其相关的属性，知识图谱展示方面使用基于 JavaScript 的 Echarts 图表库，通过引入、设置容器和绘制图谱将前文抽取的实体和关系展示在 CSCM 平台中。在招聘信息子页面中展示包含实体和关系的知识图谱，其中不同的颜色代表不同类别的实体，实体通过关系连接，通过点击鼠标触发显示实体和获取关系等相关动作。

5.3 页面展示

本文的数据来源为互联网中计算领域招聘信息，基于前文知识抽取构建计算机领域知识图谱，其中包括公司、岗位、技术领域、位置和胜任力等实体以及实体间的关系；基于收集的网页数据和文本分类算法将招聘信息分为人工智能、自然语言处理、深度学习、软件工程、数据库和操作系统六类岗位；基于 word2vec 和词共现矩阵抽取出每类岗位中所需的胜任力及其掌握程度。

综合前文的工作搭建融合知识图谱的 CSCM 平台实现基于 Elasticsearch 的职位信息全文检索和岗位胜任力图谱可视化，接下来将展示 CSCM 平台中招聘信息语义检索和岗位胜任力展示模块的相关页面。图 5.3 是 CSCM 平台的首页，其中包含平台的基本信息和关键词搜索功能。

图 5.3 CSCM 平台首页截图

5.3.1 招聘信息的语义检索

CSCM 平台同步了 Neo4j 图数据库的数据至 Elasticsearch，通过 ES 实现基于语义的全文检索，将匹配度最高的结果优先显示。招聘信息返回结果按照岗位类别被分为六类，每一个搜索结果中包含招聘名称、所属公司、公司网址、岗位标签和岗位描述等信息。部分示例如图 5.4 和图 5.5 所示。

图 5.4 招聘信息搜索页面

图 5.5 公司信息搜索页面

在招聘信息页面中，将对求职信息进行详细的展示，其中包括公司更详细的信息，例如公司领域、公司阶段和公司人数。在招聘要求中增加工作经验要求和学历要求，在岗位描述中全面的介绍招聘的要求和工作内容，具体如图 5.6 所示。

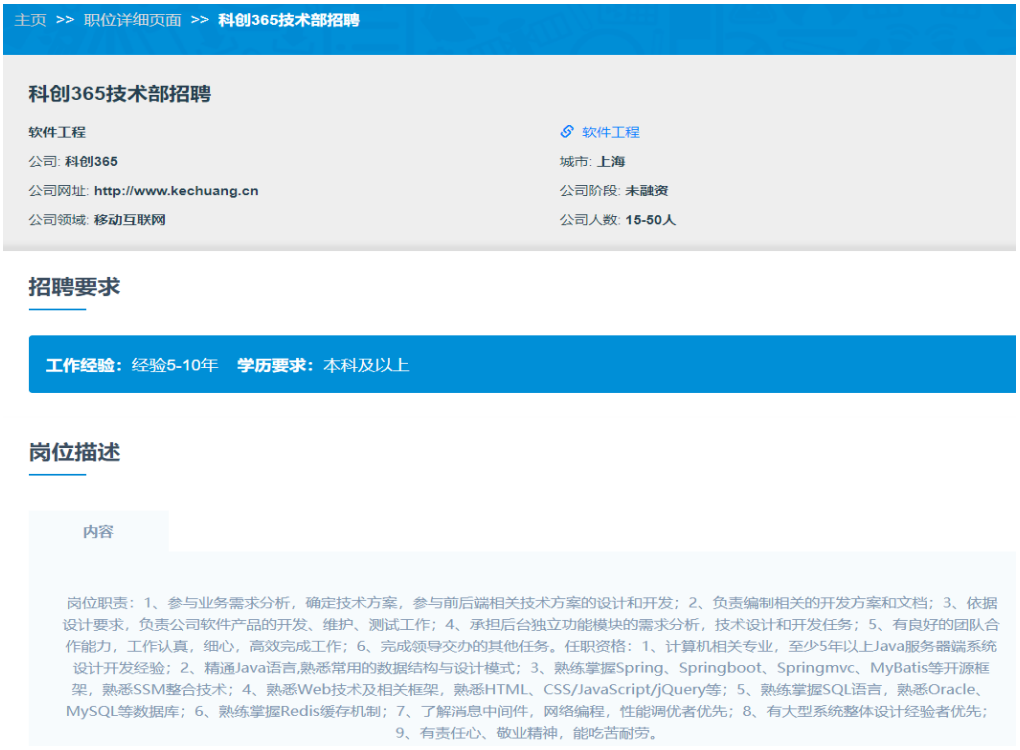

图 5.6 招聘职位详细页面

在招聘职位详情页面中还展示了职位的关键词和知识图谱。在图谱中展示当前的招聘名称及所需的胜任力、所属的岗位类别、工作地点以及所属公司等信息。关键词和知识图谱可视化是对文字性描述的总结和概括，知识图谱可视化为用户全面展示招聘的信息和与之关联的技术领域等，具体如图 5.7 所示。

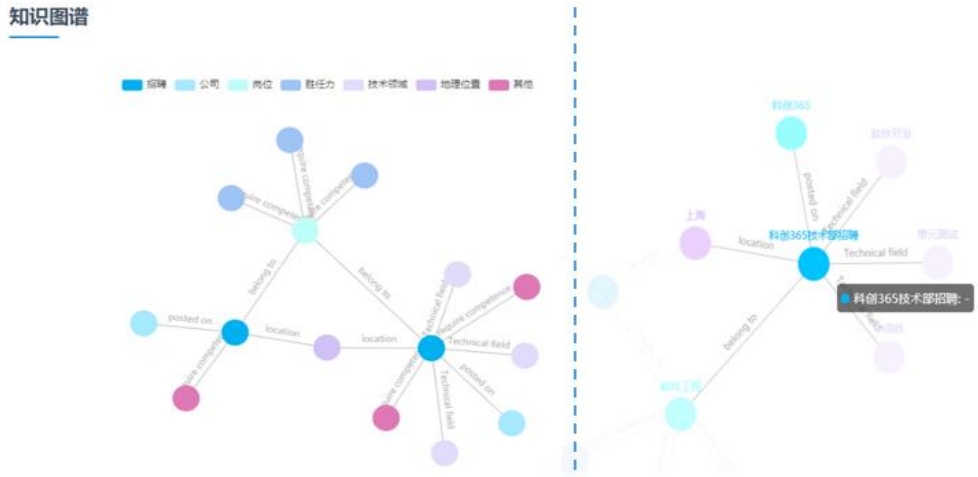

图 5.7 CSCM 平台知识图谱可视化展示图

5.3.2 胜任力需求模型的图谱展示

在岗位页面中首先通过知识图谱可视化将该类岗位所需的胜任力以及每种胜任力的程度关系展示出来，关联出此类岗位的招聘信息以及工作地点等其他实体。在所展示的掌握程度中将按照“掌握”，“熟悉”，“负责”，“了解”，“参与”分为不同关联类别，用户点击关系即可查看不同胜任力所要求的掌握程度。同时在图谱显示中可以对实体类型进行筛选，既可以查看全部图谱又可以查看当前图谱的部分子图。

通过岗位胜任力需求模型的图谱可视化功能，可以为求职者清晰地展示计算机领域不同岗位对行业中知识、技能、工具、算法等实体的掌握程度要求。在求职过程中为用户提供全面的岗位介绍和胜任力需求参考。胜任力需求模型对用户的职业生涯规划具有一定积极意义，图 5.8 为自然语言处理岗位中的胜任力展示。

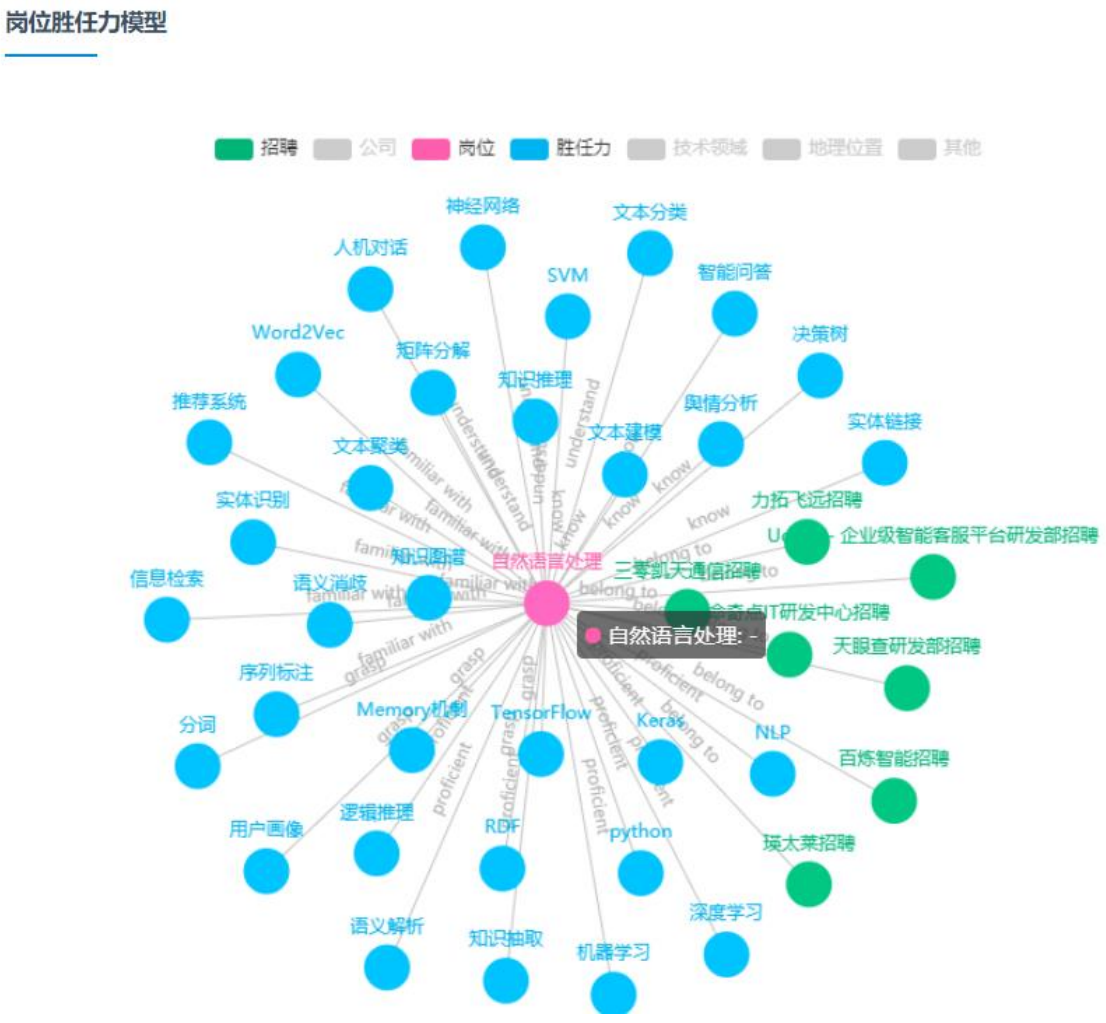

图 5.8 自然语言处理岗位胜任力展示图

在岗位胜任力需求模型的图谱展示页面中还包括对该类别的岗位信息推荐。针对具体的计算机领域的岗位检索结果按照工作地点分类显示，为用户展示不同工作地点的对应岗位的招聘信息。通过列表的方式为用户展现招聘信息、公司名称、公司网址、薪资和学历要求等详细内容，具体如图 5.9 所示。

图 5.9 岗位按照地区分类列表

5.4 本章小结

本章主要介绍计算机领域胜任力管理平台的设计与应用。首先介绍平台的整体架构，其中包括知识图谱、后端搜索引擎和前端。接着从平台设计的角度按照主要功能模块分别介绍平台，分为基于 Elasticsearch 的全文检索和基于 Vue.js 的数据展示。最后通过页面展示介绍平台的各个功能模块和效果图。

第 6 章　总结与展望

6.1 研究总结

近来，国内互联网巨头公司增设知识图谱方面实验室，工业界也将知识图谱引入到产品中提升用户体验和产品质量，由此可见知识图谱正成为工业界关注的热点。在学术界多学科交叉的研究正逐渐增多，但知识图谱与计算机领域胜任力管理相结合的研究相对较少。计算机领域岗位所需的知识和技能展示还不全面，招聘信息搜索中更多的是基于关键词而不是语义和胜任力的搜索，导致在求职过程中出现胜任力无法准确匹配或者搜索结果冗杂。

本文以计算机领域的招聘信息为数据基础，针对以上问题基于知识图谱抽取出计算机领域岗位胜任力并搭建平台为用户提供招聘信息的语义检索和岗位胜任力展示。本文通过数据收集，数据清洗、定义数据模型、构建计算机领域词典以及知识抽取和存储构建了基于计算机领域招聘信息的知识图谱。然后通过文本分类实现岗位分类，再使用共现矩阵抽取出每类岗位所需的胜任力及掌握程度。最后基于前文的工作搭建了以语义搜索和知识图谱可视化为主要功能的岗位胜任力搜索和展示平台，基于 Elasticsearch 和 Vue.js 框架实现针对公司与职位需求的语义全文检索和数据展示功能。

本文也具有一定的局限性。首先，在知识图谱构建中所定义的数据模式仍然还有很多可以改进的地方，许多与岗位胜任力模型相关的概念没有引入，实体间的关系还可以更加丰富；在知识抽取过程中对实体的抽取结果与标注的数据质量有直接关系，本文标注的数据的数量和质量还有提升的地方。其次，基于 BERT 的岗位分类模型和胜任力抽取模型仍具有改进的空间，虽然 F1 值满足了分类的基本要求，但是未来将尝试从数据和算法两方面提升模型的性能。最后，计算机领域岗位胜任力搜索与展示平台中主要是基于知识图谱的语义搜索和胜任力图谱展示，未来可引入基于知识图谱的岗位推荐和用户职业生涯规划等功能，对知识图谱在计算机领域岗位胜任力管理方面开展进一步的探索。

6.2 工作展望

本研究未来的工作将分为三部分：

（1）在知识图谱构建方面，接下来将拓展更多概念和实体节点，融合与计算机领域相关的其他学科和领域，增加数据来源进一步丰富知识图谱。例如增加与岗位胜任力模型相关的管理学科的相关概念和实体，多层次丰富知识图谱的数据模式和数据规模。

（2）在岗位胜任力抽取方面，虽然本文实现较好的岗位分类效果，但接下来将提升数据标注质量和数量实现更好的岗位分类效果；另外本文在胜任力抽取方面首先通过 word2vec 完成程度词的分类，通过词共现矩阵对胜任力进行抽取并绘制岗位胜任力图谱，下一步将对程度词类别进行扩充，争取更加详细地展示计算机领域岗位所需的胜任力及其掌握程度。

（3）在知识图谱应用方面，本文基于知识图谱实现了包括语义检索和知识图谱可视化的胜任力管理平台，但是基于知识图谱还可以提供例如知识推荐等其他的功能，未来仍有很大的探索空间。本文在完善知识图谱的基础上尝试在平台中增加岗位推荐、用户职业规划和生涯管理等其他模块，探索搭建计算机领域综合胜任力管理平台。

参考文献

[1] 王欣. 关于计算机专业人才的社会需求及计算机专业发展前景的调研分析报告[J]. 电脑知识与技术：学术交流(4 期):225-226.

[2] 刘金玲.2019 年 AI 人才缺口有多大？哪些企业在招 AI 人才？[DB/OL]. https://baijiahao.baidu.com/s?id=1626796178035528421&wfr=spider&for=pc

[3] 李晓明. 关于计算机人才需求的调研报告[J]. 计算机教育, No.(08):12-19.

[4] 李芳玲. 面向用户的招聘类网站评价研究[D]. 2016.

[5] 薛琴. 论基于岗位胜任力模型的人才选拔[J]. 江苏商论, 2007(9):113-115.

[6] Singhal A. Introducing the knowledge graph: things, not strings[J]. Official google blog, 2012.

[7] 陈悦, 刘则渊. 悄然兴起的科学知识图谱%The rise of mapping knowledge domain[J]. 科学学研究, 2005, 023(002):149-154.

[8] Bollacker K, Evans C, Paritosh P, et al. Freebase: a collaboratively created graph database for structuring human knowledge[C]// Proceedings of the 2008 ACM SIGMOD international conference on Management of data. ACM, 2008: 1247-1250.

[9] Auer S, Bizer C, Kobilarov G, et al. Dbpedia: A nucleus for a web of open data[C]// Proceedings of the semantic web. Springer, Berlin, Heidelberg, 2017:722-735.

[10] Tang J, Zhang J, Yao L, et al. ArnetMiner: extraction and mining of academic social networks[C]// ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. DBLP, 2008:990-998.

[11] AceKGWang R . AceKG: A Large-scale Knowledge Graph for Academic Data Mining[J]. 2018.[11] AceKGWang R . AceKG: A Large-scale Knowledge Graph for Academic Data Mining[J]. 2018.

[12] Noyons E C M , Calero-Medina C . Applying bibliometric mapping in a high level science policy context[J]. Scientometrics, 2009, 79(2):261-275.

[13] 薛琴. 论基于岗位胜任力模型的人才选拔[J]. 江苏商论, 2007(9):113-115.

[14] McClelland, David C . Testing for competence rather than for "intelligence."[J]. American Psychologist, 1973, 28(1):1-14.

[15] clagan P A . Competencies: The next generation[J]. Training & Development, 1997, 51(5):83.

[16] Miranda S , Orciuoli F , Loia V , et al. An ontology-based model for competence management[J]. Data & Knowledge Engineering, 2017, 107:51-66.

[17] 戴成秋, 冯君莹. 基于岗位胜任力的新建本科院校计算机专业课程体系研究[J]. 科技创新导报, 2014(28):248-249.

[18] 纪威. 基于胜任力模型的计算机专业创新人才培养研究[J]. 计算机与网络, 2015, v.41;No.517(21):73-76.

[19] https://blog.csdn.net/fufu_good/article/details/104189073

[20] Berners-Lee, Tim, Handler, James, Lassila, Ora. The Semantic Web[J]. Scientific American, 2006, 284(5):34--43.

[21] 刘峤,李杨,段宏,刘瑶,秦志光.知识图谱构建技术综述[J].计算机研究与发展,2016,53(03):582-600.

[22] Broekstra J , Kampman A , Harmelen F V . Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema[J]. 2002.

[23] Verborgh, Ruben, Vander Sande, Miel, Hartig, Olaf,等. Triple Pattern Fragments: A low-cost knowledge graph interface for the Web[J]. Web Semantics Science Services & Agents on the World Wide Web:S1570826816000214.

[24] Nadeau D , Sekine S . A survey of named entity recognition and classification[J]. Lingvisticae Investigationes, 2007, 30(1):3-26.

[25] Fresko M , Rosenfeld B , Feldman R . A hybrid approach to NER by MEMM and manual rules[C]// Acm International Conference on Information & Knowledge Management. ACM, 2005.

[26] Song Yu, Kim Eunju, Lee Gary Geunbae. POSBIOTM—NER: a trainable biomedical named-entity recognition system[J]. Bioinformatics, 2005(11):11.

[27] A. McCallum, D. Freitag, and F. Pereira, “Maximum entropy Markov models for

information extraction and segmentation," in ICML, pp. 591–598, 2000.

[28] Brereton R G , Lloyd G R . Support Vector Machines for classification and regression[J]. ANALYST, 2010, 135(2):230-0.

[29] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In International Conference on Machine Learning (ICML), 2001.

[30] Pengda Qin, Weiran Xu, William Yang Wang. Robust Distant Supervision Relation Extraction via Deep Reinforcement Learning[C]// Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2018.

[31] Jun Feng, Minlie Huang, Li Zhao, Yang Yang, Xiaoyan Zhu. Reinforcement Learning for Relation Classification from Noisy Data. The 32th AAAI Conference on Artificial Intelligence (AAAI 2018).

[32] O. Lassila and R. Swick. Resource Description Framework (RDF) Model and Syntax Specification. http://www.w3.org/TR/REC-rdf-syntax/, 1998.

[33] Horrocks I , Patel-Schneider P F , Harmelen F V . From SHIQ and RDF to OWL: the making of a web ontology language[J]. web semantics science services & agents on the world wide web, 2004, 1(1):7-26.

[34] LyleM． Spencer， SigeM． Spencer． ComptenceatWork: Modelsfor-SuperiorPerformance［M］． NewYork: JohnWiley ＆Sons，Inc，1993．

[35] Karoline, Koeppen, Johannes, et al. Current Issues in Competence Modeling and Assessment[J].

[36] 姜海燕．岗位胜任力评价研究［D］．南京：河海大学，2005.

[37] Alldredge M E , Nilan K J . 3M's leadership competency model: An internally developed solution[J]. Human Resource Management, 2000, 39(2‐3):133-145.

[38] Selmer J , Chiu R . Required human resources competencies in the future: a framework for developing HR executives in Hong Kong[J]. Journal of World Business, 2004, 39(4):324-336.

[39] Schmidt A , Kunzmann C . Towards a Human Resource Development Ontology for Combining Competence Management and Technology-Enhanced Workplace Learning[C]// On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops, OTM Confederated International Workshops and Posters, AWeSOMe, CAMS, COMINF, IS, KSinBIT, MIOS-CIAO, MONET, OnToContent, ORM, PerSys, OTM Academy Doctoral Consortium, RDDS, SWWS, and SeBG. Springer-Verlag, 2006.

[40] 陈小中. 基于岗位胜任力的高职计算机网络专业人才培养模式研究[J]. 高教学刊, 2016, 000(006):66-67,69.

[41] 马郁,郭磊.以岗位胜任力为导向的中职计算机专业学生能力培养体系的研究与实践[J].电脑迷,2018(11):262.

[42] Niculescu, C., Trăuşan-Matu, S. (2009), An Ontology-centered Approach for Designing an Interactive Competence Management System for IT Companies; Informatica Economică, vol. 13, no. 4;

[43] Kimble C , De Vasconcelos J B , Rocha A . Competence management in knowledge intensive organizations using consensual knowledge and ontologies[J]. Information systems frontiers, 2016, 18(6):1119-1130.

[44] Hochreiter, S, Schmidhuber, J. Long Short-Term Memory[J]. Neural Computation, 9(8):1735-1780.

[45] Lample G , Ballesteros M , Subramanian S , et al. Neural Architectures for Named Entity Recognition[J]. 2016.

[46] Devlin J , Chang M W , Lee K , et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding[J]. 2018.

[47] Liu W , Zhou P , Zhao Z , et al. K-BERT: Enabling Language Representation with Knowledge Graph[J]. 2019.

[48] Mikolov, Tomas, Sutskever, Ilya, Chen, Kai,等. Distributed Representations of Words and Phrases and their Compositionality[J]. Advances in Neural Information Processing Systems, 2013, 26:3111-3119.

[49] Wang Ling, Chris Dyer, Alan Black, and Isabel Trancoso. 2015. Two/too simple adaptations of word2vec for syntax problems. In Proc. NAACL.

[50] https://www.cnblogs.com/pinard/p/7243513.html

[51] Su, J. L.: A Hierarchical Relation Extraction Model with Pointer-Tagging Hybrid Structure. Github, https://github.com/bojone/kg-2019 (2019)

[52] Zhang C L , Luo J H , Wei X S , et al. In Defense of Fully Connected Layers in Visual Representation Transfer[J]. 2017.

[53] Wang H, Wang H, Wen J R, et al. An Inference Approach to Basic Level of Categorization[C].ACM International on Conference on Information and Knowledge Management. ACM, 2015:653-662

致 谢

岁月如梭，在吉林大学攻读硕士研究生的三年似乎转瞬即逝。回顾一路以来的学习历程，自己掌握了许多专业知识和技能，提升了综合能力。在即将毕业的时刻内心充满感恩，我想感谢所有帮助过我的老师、同学、家人和朋友，因为有你们才让我的研究生学习时光充实而有意义。

首先我最想感谢的是我的导师徐昊教授，在学术研究方面导师给予我很多建议和学习资源，教导我做学术研究应当具有怎样的思维逻辑和科研方法。生活中的导师更像是一位兄长，在导师的鼓励下我成功申请到欧盟交流的机会，这对我来说意义非凡。在我求职过程中导师耐心地指导我并给予很多中肯的建议和无私的帮助。三年的学习生活中，导师传授的理念和处事方法对我影响非常大，这在日后也将使我受益匪浅。最后我怀着十分感激的心情向徐昊教授说一声，谢谢您！

接着我想感谢 Human AI 实验室一起学习的小伙伴们。感谢所有同学在我研究生期间对我无私的帮助，感谢你们在讨论班中分享科技前沿论文，感谢你们让我处在学术氛围如此浓厚的环境中。感谢数字大学项目让我深入了解知识图谱并且收获满满，那些和小伙伴一起努力的时光让我感到充实而快乐。在完成毕业设计期间，感谢宋瑞、迟杨和晓慧对我的帮助。最后祝愿实验室所有同学拥有锦绣前程，奔向美好的未来。

最后我想感谢我的家人和朋友，感谢你们在我做出考研选择的时候无条件的支持我，这让我更加勇敢的追逐自己的梦想。感谢家人一直以来为我提供的物质和精神支持，我将怀着感恩的心回报你们多年的养育之恩。

在吉林大学的七年求学时光充实而有意义，我在这里收获了导师的教导、专业的知识、综合的能力和同学间的友情，这将成为前行的铠甲让我在未来的路上勇往直前。