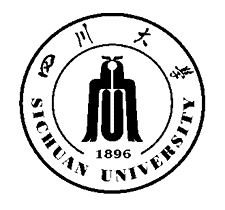


**本科生毕业论文（设计）**

****

**题 目 基于知识图谱的证券类问答系统设计**

**学 院 计算机学院**

**专 业 计算金融交叉试验班**

**学生姓名 杜林峰**

**学 号** 2019141460185 **年级** 2019

**指导教师 段磊**

**教务处制表**

**二〇二三年四月二十日**

基于知识图谱的证券类问答系统设计

专业：计算金融交叉试验班

学生：杜林峰 指导教师：段磊

摘 要 近年来，随着科技的发展，在人类生活的方方面面都可以看到互联网技术的身影，互联网的普及，让人们可以获取日常生活中需要的各种信息，提高了人民的生活和工作效率。与此同时，互联网信息的几何级数增长也给人们快速获取信息带来了巨大的麻烦。应运而生的搜索引擎给用户提供了便利的知识获取渠道，在某种程度上，让用户对信息的检索更加方便。对于面向特定领域的相关问题，传统的搜索引擎在技术上无法做到返回精准简洁的答案，仍需要用户对答案进行甄别，或向专业客服人员进行询问，这无疑浪费了用户和专业人员的时间。大数据技术的发展和人工智能技术的深入研究，使得问答系统的出现成为可能。很多公司和研究机构开始进行深入研究，相继推出问答系统产品，比如百度公司的小度机器人、科大讯飞的 AIUI平台、微软的小冰和小娜等。但在金融领域的相关产品几乎为零，金融领域的“铁粉”对于金融知识的获取依然举步维艰。

本文针对金融领域中的证券类产品，对金融问答系统和相关技术做了深入研究，并设计了一个基于

知识图谱的证券类问答系统，主要开展的工作如下。

（1）采用自底向上的方式构建金融知识图谱。首先要获取数据，通过接口调用和设计爬虫程序获得数据，并将数据进行清洗。之后依照一些规则对得到的数据进行实体，属性和关系的提取，转化为三元组。最后通过neo4j图数据库将获得的三元组信息进行存储，实现可视化查询。

（2）基于知识图谱的金融问答算法设计。算法的核心是将用户所提问题中的关键字链接到知识图谱中，返回直观简洁的答案。这一过程主要包括三个任务，训练文本分类模型，对金融问句的分类以及答案查询。首先基于今日头条的数据，使用Fasttext算法训练文本分类模型，当用户问题被分类为闲聊类时，会通过闲聊接口给用户作出回答。当问题被分类为领域专业问题时，会根据关键词检索常见问题库，通过关键词匹配，对问句进行分类，证券类问题本身属于封闭域类场景，对领域问题进行穷举并分类，然后使用cypher的match去匹配查找neo4j，根据返回数据组装问句回答，最后返回结果。同时设置了问题分类实验，在测试的数据集上，训练出的模型取得了0.9 的F1值。

（3）金融问答系统搭建。使用Django框架搭建基于Web端的证券类问答系统，整个系统分为三个模块，数据模块、问答模块、前端展示模块。实验测试结果表明，本论文的问答系统科学有效。

关键词 问答系统；证券类领域；知识图谱；文本匹配；文本分类；

**Design of securities financial Question and answer System based on Knowledge graph**

Major: Computational Finance

Student: DU Linfeng Adviser: DUAN Lei

**Abstract** In recent years, with the development of science and technology, Internet technology can be

seen in all aspects of human life. The popularization of the Internet enables people to obtain all kinds of information needed in daily life, and improves people's life and work efficiency. At the same time, the geometric growth of Internet information also brings great trouble to people to get information quickly. The search engine emerges at the historic moment to provide users with convenient access to knowledge channels, to a certain extent, make users more convenient to retrieve information. For specific domain related questions, the traditional search engine is technically unable to return accurate and concise answers, but users still need to screen the answers or ask professional customer service staff, which undoubtedly wastes the time of users and professionals. The development of big data technology and the in-depth study of artificial intelligence technology make the emergence of question and answer system possible. Many companies and research institutions began to carry out in-depth research and successively launched question and answer system products, such as Baidu's Xiaodu Robot, iFlytek's AIUI platform, Microsoft's Xiaoice and Xiaona, etc. However, there are almost no related products in the financial field, and the "die-hard fans" in the financial field are still struggling to acquire financial knowledge.

In this paper, the securities products in the financial field, the financial question and answer system and related technologies to do in-depth research, and design a based on

The main work of the securities question and answer system of Knowledge graph is as follows.

(1) A bottom-up approach is adopted to construct a financial knowledge map. First of all, we need to obtain data, obtain data through interface call and design crawler program, and clean the data. After that, entities, attributes and relationships are extracted and converted into triples according to some rules. Finally, the triplet information obtained is stored through the neo4j graph database to realize visual query.

(2) Design of financial question answering algorithm based on knowledge graph. The core of the algorithm is to link the key words in the questions raised by users to the knowledge graph, and return intuitive and concise answers. This process mainly includes three tasks: training text classification model, classification of financial questions and answer query. First, based on the data of Toutiao, the Fasttext algorithm is used to train the text classification model. When a user's question is classified as chatty, it will answer the user through the chatty interface. When questions are classified as domain professional questions, the frequently asked questions database will be searched according to keywords, and questions will be classified through keyword matching. Securities questions themselves belong to closed domain scenarios, domain questions will be exhaustive and classified, and then cypher match will be used to find neo4j for matching, and question answers will be assembled according to the returned data. Finally, the result is returned. At the same time, the problem classification experiment was set up. On the test data set, the trained model obtained the F1 value of 0.9.

(3) Construction of financial question and answer system. Using Django framework to build a securities question and answer system based on Web end, the whole system is divided into three modules, data module, question and answer module, front-end display module. Experimental test results show that the question answering system of this paper is scientific and effective.

**Key words** Question and answer system; Securities field; Knowledge map; Text matching; Text classification;

**目 录**

[第1章 绪论 1](#_Toc102997270)

[1.1 研究背景及意义 1](#_Toc102997271)

[1.2 主要研究工作 2](#_Toc102997272)

[1.3 本文组织结构 3](#_Toc102997273)

[第2章 相关工作 5](#_Toc102997274)

[2.1 基于传统统计学与机器学习方法的信贷风控建模技术研究 5](#_Toc102997275)

[2.1.1 基于监督学习的信贷风控建模 5](#_Toc102997276)

[2.1.2 基于无监督学习的信贷风控建模 6](#_Toc102997277)

[2.2 联邦学习相关技术研究 6](#_Toc102997278)

[2.2.1 联邦学习基本概念及分类 6](#_Toc102997279)

[2.2.2 联邦学习训练步骤 9](#_Toc102997280)

[2.2.3 联邦学习前沿技术研究进展 10](#_Toc102997281)

[2.2.4 联邦学习前沿应用场景 15](#_Toc102997282)

[2.3 本章小结 16](#_Toc102997283)

[第3章 信贷风控系统的分析与设计 17](#_Toc102997284)

[3.1 问题背景 17](#_Toc102997285)

[3.2 问题定义 18](#_Toc102997286)

[3.2 信贷风控系统重难点分析 18](#_Toc102997287)

[3.3 信贷风控系统各模块分析及总体设计 19](#_Toc102997288)

[3.3.1 数据预处理模块 20](#_Toc102997289)

[3.3.2 客户端模块 21](#_Toc102997290)

[3.3.3 中心服务器模块 22](#_Toc102997291)

[3.4 本章小结 22](#_Toc102997292)

[第4章 信贷风控系统的构建与实验 23](#_Toc102997293)

[4.1 系统开发环境 23](#_Toc102997294)

[4.1.1 开发工具 23](#_Toc102997295)

[4.1.2 开发语言及第三方框架/库/包 23](#_Toc102997296)

[4.2 信贷风控系统各重点模块的实现 25](#_Toc102997297)

[4.2.1 数据预处理模块 25](#_Toc102997298)

[4.2.2 客户端模块 27](#_Toc102997299)

[4.2.3 中心服务器模块 28](#_Toc102997300)

[4.2.4 本节常用符号汇总 32](#_Toc102997301)

[4.3 实验与结果分析 32](#_Toc102997302)

[4.3.1 实验环境 32](#_Toc102997303)

[4.3.2 实验数据分析 33](#_Toc102997304)

[4.3.3 实验评价指标 33](#_Toc102997305)

[4.3.4 实验结果分析 34](#_Toc102997306)

[4.4 本章小结 43](#_Toc102997307)

[第5章 总结与展望 44](#_Toc102997308)

[5.1 课题工作总结 44](#_Toc102997309)

[5.2 未来工作展望 44](#_Toc102997310)

[参考文献 46](#_Toc102997311)

[声明 50](#_Toc102997312)

[致谢 51](#_Toc102997313)

**图目录**

[图2-1 横向联邦学习 7](#_Toc101876884)

[图2-2 纵向联邦学习 8](#_Toc101876885)

[图2-3 联邦迁移学习 8](#_Toc101876886)

[图2-4 联邦学习典型训练步骤 9](#_Toc101876887)

[图3-1 基于联邦学习的信贷风控系统组织结构图 19](#_Toc101876888)

[图4-1 数据处理模块流程图 25](#_Toc101876889)

[图4-2 BorderlineSMOTE方法示意图 26](#_Toc101876890)

[图4-3 不同过采样比例下的联邦建模指标对比 38](#_Toc101876891)

[图4-4 不同客户端数量下的指标对比 40](#_Toc101876892)

[图4-5 FedAvgOS方法实验结果 42](#_Toc101876893)

**表目录**

[表3-1 本节符号汇总 18](#_Toc101876988)

[表4-1 系统开发环境 23](#_Toc101876989)

[表4-2 本节符号汇总 32](#_Toc101876990)

[表4-3 数据集信息表 33](#_Toc101876991)

[表4-4 混淆矩阵 33](#_Toc101876992)

[表4-5 本地客户端模型参数设置 35](#_Toc101876993)

[表4-6 本地客户端模型在不同过采样方法下的结果 36](#_Toc101876994)

[表4-7 本地客户端模型在不同过采样比例下的结果 37](#_Toc101876995)

[表4-8 不同参数聚合方法下的实验数据对比 41](#_Toc101876996)

**第1章 绪论**

1.1 研究背景及意义

近年来，随着脱贫攻坚取得圆满成功，人民生活欣欣向荣，越来越多的人开始或已经去了解金融领域的方方面面，与此同时人工智能也推动着金融行业的发展。网络上充斥着大批的金融领域相关信息，用户有着急需对金融领域进行全面了解的诉求，基于内容的搜索引擎无法直接给出明确的答案。用户通常会询问专业的业务人员，而往往由于业务人员人数有限、时间有限、精力有限，无法为用户在第一时间给用户做出解答，更无法做到全天不间断给用户提供服务。因此金融领域把深度学习和问答系统结合，构建金融问答系统具有重要的理论研究意义和使用价值。

问答系统是一种通过理解用户自然语言问句，并可以从大量数据中找出简洁明了的答案的检索系统。因为中英文在语法和句法等结构上有很大的差异，中文是一种意合语言，词性转换无变化、动词没有单复数的区别、没有复杂的语法规则，对于中文句子的理解变得更加困难，不能直接将国外的科研成果运用到中文。另一方面，因为缺少成体系的中文语料库、知识库等，使得中文问答系统起步较晚，发展较为缓慢。

（1）国内研究现状

国内问答系统领域的科研机构包括研究所、高等院校和互联网公司，研究成果被广泛运用。中科院计算技术研究所开发的NKI问答系统，以国家知识基础设施建设为背景集成了各个学科的知识，同时融入了各个学科领域专家的个人知识，建成了强大的知识库，可以为各种领域的公共知识进行问答服务。北京大学研发的中文问答系统gAnswer，是基于PKU知识库，面向开放域的自然语言问答系统，它可以将问句通过依存句法分析，转变为查询图，该查询图由于蕴含语义信息被称为语义查询图，然后语义查询图会被转化为SPARQL查询语句，最后在图数据库中执行该查询语句，将问题答案反馈给用户。台湾国防大学研发的CQAS问答系统，重点进行了命名实体和关系处理的科研。它将中文文本转换为三元组文本，通过用中文的文本特征来表示中文文本，增强现有搜索引擎的体系结构来支持中文自然语言问答。科大讯飞AI实验室于2015年研发的科大讯飞AIUI平台，可以通过设备或者应用接入AIUI，实现播放音视频、查询天气、对智能家居进行控制等能力，已被成功投放使用如海底捞的智能客服，它通过接入讯飞AIUI技术，使得用户可以直接使用移动电话与人工智能客服终端进行交互，同时可以完成订餐预约、点餐等服务。据统计，截止2018年全国约150家门店都已上线人工智能客服，服务占比约为85%，平均每家门店至少减少1名人工客服人员，极大节约了人力成本，公开数据显示，其每年可节省成本达1000万元。此外，小度机器人——百度AI的形象代言人，在国内某档栏目以一鸣惊人之势进入众人视野，它是百度公司研发的一种问答机器人，能够实现问答咨询、自然情感对话和引领讲解等功能，其中它的问答咨询功能的核心来自于百度深度问答系统，深度问答系统基于深度问答技术，它能够深度理解人类语言，可以做到用人类的思维理解语言的含义，同时基于百度强大的搜索引擎坐拥海量网络数据，可以提供在海量网页中进行搜索的技术支持。语义深度理解基于百度的自然语言处理技术，通过该技术对问题进行深度理解并了解用户意图，通过海量数据的查询和语义的理解，整合出答案信息。该问答系统对于百科类知识尤为擅长，上到宇宙太空，下到江河湖海、中到乐理影视，博古通今满腹经纶，同时他还可以处理原因类、方法类和观点类等问题，可以直面问题，高效给出准确答案。

（2）国外研究现状

在二十世纪五十年代，计算机之父图灵第一次提出"机器智能"，同时提出了测试机器是否拥有智能的一种方法————“图灵测试”，随后麻省理工于1966年研发出了ELIZA，作为最早的问答系统，由于不具有完善的知识库，只能算是简单的问答系统，它通过分析人类提问内容，找出关键词，以做出相应的回答，整个程序代码只有200行左右，这种问答的模式并非是理解人类的意图和想法，而是通过分析所键入的文字，根据特定字句的组合，生成新的字句组合，同时加入了对话引导，从而让人类以为对话者是真人。与人脑有额外的情感不同，该系统通过分析大量数据挖掘人类相关偏好，达到问答的目的。1973年，Lunar系统问世，用于回答阿波罗探月采集的样本相关的问题，同期另一个人工智能专家系统SHRDLE进入人们视野，该系统可以面向玩具世界，回答其形状和状态问题。1995年，Androutsopoulos（Androutsopoulos等，1995）等人针对基于数据库的问答系统涉及的问题、方法和进行了总结，基于检索的问答系统开始兴起，麻省理工开发的START系统以及密歇根大学研发的Answerbus问答系统都是这个时代的产物。START问答系统能够回答多种学科的常识性问题，例如历史、地理、科技和娱乐等方面，对常识性问题直接给出答案，并且不相关的信息在答案中一般不会出现。Answerbus基于网页信息检索技术，是一种句子级问答系统，它支持英语问答、葡萄牙语问答、意大利语问答、西班牙语问答以及法语问答，通过多个搜索引擎进行检索与句子相关的资源。同样它支持多个领域，基于问题从网页中抽取相关的内容，准确性和响应速度在当时是其他同类系统不能媲美的。

2011年，IBM研发的深度问答系统沃森（Waston）在美国一档竞答节目中打败了卫冕冠军拔得头筹，成为人工智能历史上具有里程碑意义的智能问答系统。沃森会通过图像识别技术识别屏幕的文字，将文字中的基本信息比如时间、人名地名等抽取出来，和知识库进行比对，这个比对的过程是将搜索任务拆成很多小的子搜索并行处理，将比对结果打上分类标签，然后通过提取文本的语义结构信息，例如词性，依存句法等来分析问题。其中一般每一步都会用分治的思想转化为求解子问题，一个只有数十个词的问题可能会被分解为成千上万个子问题。此外，当问题难以解决时，为了得到答案，沃森会用一定的策略引入新的问题。最后会使用大量的算法对相关的答案进行打分评估，其中会涉及到答案的类别、语法结构、词性等是否合理，最后会将整理好的最终答案转换成语音反馈给用户。沃森的问世让问答在商业界和科研界变得炙手可热，同时推动了更多优秀问答系统的发展。后来陆续诞生了许多基于设备的应用型问答系统，比如苹果的Siri助手，微软的小娜助手，谷歌的Google Now。

（3）研究现状总结

从本课题目前搜集到的资料来看，虽然目前国内外在问答系统的研究已经相当的成熟与智能，但在这些研究中也存在着不足的地方。比如，当前金融类问答系统没有针对证券类产品单个领域所做的，其不能满足用户在证券类产品购买使的咨询需求；另外，如今代表问答系统顶尖水平的ChatGPT在很多专业领域方面效果不如传统的KBQA，这给本文的写作提供了一些空问和可能性，本文计划针对这一问题述行相应的拓展和研究。

1.2 主要研究工作

本文以金融项目为应用背景，运用现阶段流行的检索式问答和模板匹配式问答相关技术，将NLP技术与之结合，应用到证券类领域的问答服务中。本文将在构建的知识图谱的基础上，设计并实现一个证券类问答系统。本文的问答系统一方面可以为用户提供精准的问答服务，另一方面可以减轻专业人员的工作量。具体研究内容如下：

（1）知识图谱设计。图谱的构建主要分为三个部分：知识采集，知识抽取，知识存储和可视化。人工智能，数据先行。无论是系统开发和模型训练，都离不开数据的支撑。

在金融领域，因缺乏数据信息，对于问答系统方向还没有较好的落地应用，本文的数据主要来源于tushare以及各大财经网站，编写Python程序获取到数据，对数据进行预处理后，按照一定的方法进行实体，属性，关系的抽取。采用自底向上的方式构建证券类领域的知识图谱，最后利用neo4j图数据库进行存储，为后续的问答系统相关模型训练和开发打下基础。

（2）基于知识图谱的问答关键技术研究。核心是将用户所提问题中的关键字链接到知识图谱中，返回直观简洁的答案。这一过程主要包括两个部分：问句解析和答案检索。本文首先在构建的知识图谱的基础上，根据常见的用户问题，构建查询模板，其次再对今日头条的数据进行训练，使其能够对用户的问题进行准确分类，当问题被分为金融领域问题时，再通过设定好的模板进行匹配，若分类为其他领域或者匹配失败则返回闲聊语句，最后对查询结果，对关键信息进行填充，转化为cypher语句，完成知识图谱的查询和推理，实现金融智能问答。

（3）证券类问答系统搭建。使用Django框架搭建基于Web端的金融问答系统，为用户提供证券类产品咨询平台。系统分为三个模块，数据模块、逻辑模块、前端展示模块。数据模块为证券类问答系统提供数据支撑，逻辑模块主要包括问答流程的相关算法并为前端展示模块提供对外服务API接口，前端展示模块主要用于实现系统和用户的交互。

1.3 本文组织结构

本文共分为六章，具体内容如下：

第1章 绪论。首先介绍了本文的研究背景，在研究背景基础上提出了研究的意义，然后对国内外知识图谱、基于知识图谱问答的研究现状进行了分析，最后对本文的研究内容和整体结构进行了简单介绍。

第2章 相关技术。主要介绍与本文研究内容有关的基础理论和关键技术，首先介绍了知识图谱的特点，与构建方法；然后介绍了自动问答的几个常用方法。

第3章 证券类领域知识图谱的构建。首先介绍了知识图谱的构建流程，然后从知识采集、知识抽取及知识存储和查询这四个部分进行了详细的阐述，最后展示了知识图谱的查询界面。

第4章 基于知识图谱的问答方法实现。(**重点**)

第5章 证券类问答系统的设计与实现。首先介绍了问答系统的运行环境，然后介绍了系统的三个模块，包括数据模块、逻辑模块和前端展示模块，并对每个模块进行了具体介绍，最后对系统进行了测试与演示。

第6章 总结与展望。总结全文的相关工作，分析了论文的不足之处及未来发展方向。

2.1 知识图谱构建方法

知识图谱主要由若干实体对和关系构成的三元组<ex，r，e2>组成，ex、ex 分别为实体，r表示实体之间的关系，如“<中国移动，股票代码，00941>”。知识图谱的构建主要分为三个过程：实体识别、关系抽取和知识融合。

2.1.1 实体识别

知识图谱构建中的实体识别和自动问答方法中的实体识别任务一致，从非结构化的文本中识别出蕴含信息的特定类型的实体，例如金融领域里的公司、人物、产品、销售额、日期等。实体识别任务采用精确匹配的原则来判断一个实体能否被正确地识别出来，识别过程包括实体边界识别和实体类型确定两部分，即实体的描述边界判断正确且实体的类型标记正确。实体识别的错误类型主要有三种，一种是文本的边界预测错误但其包含的实体词和实体类型标记是正确的，一种是文本边界预测正确但实体类型标记错误，还有一种是实体的边界和实体的类型全部标记错误。

实体识别通常采用以下几种方法：（1）基于规则的方法：利用人工编写规则，通过文本与规则的匹配来识别出实体词；（2）基于机器学习结合特征模板的方法：NER任务可以通过机器学习的方法转化成序列标注问题。常用的模型包括隐马尔科夫模型（HMM）、条件随机场（CRF）等，通过定义特征模板结合CRF模型进行序列标注的训练，通过维特比解码来得到最优的标记序列：（3）基于神经网络的方法：同样将NER作为一个序列标注任务处理，首先将文本转换为对应的语义向量，输入到某个神经网络中（如BERT、LSTM等），用神经网络学习自动提取特征，最后通过接入CRF模型层来完成序列标注。在具体研究中，王宁等《通过规则及知识库匹配方式实现了金融新闻中的公司名抽取，在金融领域实体识别方面有不错表现；IIuang等l--提出使用双向长短期记忆网络（Bi-LSTM）接入条件随机场（CRF）的方式实现句子级别的标签预测，在命名实体识别任务上取得了不错的效果，被广泛应用。

2.1.2 关系抽取

关系抽取（Relation Extraction.RE）主要目标是从给定的一段自然语言文本之中，抽取出目标实体之间的关系。关系抽取任务通常置于实体识别任务之后，实体识别将文本中的实体信息抽取后，实体识别方法需要围绕给定实体，在文本信息中捕获实体之间蕴含的关系，本质上，RE任务是一个多分类任务。关系抽取技术被广泛应用于构建知识图谱任务中。目前解决实体关系抽取问题可以分为流水线（Pipelined）方法和联合抽取（Joint）两种方法。

流水线方法指的是首先通过实体识别提取出实体，然后再识别这些实体之间的关系，这种方式是一个分离的框架，使得实体识别与关系分类两个子任务，易于处理且更加灵活；联合抽取则将模型应用于对实体识别和实体关系抽取任务中进行联合学习。与流水线方法不同，联合学习模型是利用单个神经网络的端到端模型同时实现实体识别和关系抽取任务，将实体与关系一起抽取出来。

2.1.3 关系抽取

当知识图谱的数据来源不唯一时，多个知识源内可能包含同一实体描述因此需要进行知识融合，将不同来源的相同实体节点合并为同一节点，使知识图谱更符合现实逻辑关系。

知识融合过程主要包含实体融合（实体对齐）和属性融合，属性独合的工作较为简单，主要是将同意义属性进行合并，同类型的属性进行语法和数值的规范，如日期格式、金融单位等等。

实体融合则相对较为复杂，主要目标是判断不同数据源中的2个实体是否指向了真实世界中的同一对象，将相同的对象进行合并，降低知识图谱的冗余度，使知识图谱的实体描述更加准确全面。实体融合的最大挑战是如何判断不同数据源的两个实体实际上在描述同一对象。在学术研究中，通常采用“实体相似度”来量化两个实体间的相似程度，当实体相似度越大，则两个实体更接近于描述同一现实对象；反之，实体相似度越小，则两个实体越独立，描述同一现实对象的可能性越小。近年来关于实体对齐的研究也着眼于如何设计更合理的实体相似度算法，使实体相似度描述更为准确。

现有的实体对齐算法可分为两类：一类是基于概率模型的实体对齐方法另一类是基于机器学习的实体对齐方法。基于传统概率模型对齐方法着重考虑实体对应属性的相似性，主要通过属性的相似程度对实体对进行评分，将实体匹配问题转变为一种分类问题，目前大量的实体对齐相关研究都以此方法为基础，本文在知识融合过程中也借鉴了这种对齐方法。该方法的直观思路是加和两个实体所有匹配属性的相似度评分，之后设置2个阈值Slaw和Shigh，当总相似度评分sim≥5h1gh时，认为两个实体匹配：当评分stow<sim<sugh时，认为两个实体可能匹配，此时需要人工加以判断；当评分sim<stow时，则认为两个实体不匹配。属性相似度可以由编辑距离或余弦相似度计算得出，不同属性的权重也可以通过 TF-IDF 等方式统计计算，具体计算方式可以根据数据集的具体情况进行设计；基于机器学习的实体对齐方法则将实体对齐视为二分类问题，通过预先标注部分实体对作为训练数据，训练分类模型，之后使用模型对实际数据进行预测分类。

**第2章 相关技术**

2.2 基于知识图谱的自动问答方法

基于知识图谱的问答系统的主要特征是知识图谱，系统依赖一个或多个领域的实体，并基于图谱进行推理或演绎，深度回答用户的问题，基于知识图谱的问答系统更擅长回答知识性问题，与基于模板的聊天机器人有所不同的是它更直接、直观的给用户答案。对于不能回答、或不知道的问题，一般直接返回失败，而不是转移话题避免尴尬。

2.2.1 基于模板匹配的问答方法

基于模板匹配的问答方法属于比较传统的自动问答方法。该方法通过提前构建一组模板参数，形成问题的查询表达式，对问题文本进行规则匹配。具体来说，基于模板匹配的问答方法使用建立好的模板对文本内容进行匹配，对于匹配的问题，通过模板预设的映射格式，将问题文本中的对应内容映射到相应的变量中，例如对于模板"A的B是多少"可以匹配问题"中国移动的股票代码是多少"，此时"中国移动"被赋予变量A，"股票代码"被赋予变量B，之后按照模板映射的数据查询语言构造对应的查询表达式，完成数据查询，返回问题答案。

可以看出，基于规则模板的方法不涉及对语句语义的分析，只需要通过预设的问题模板对问题文本逐一匹配，即可抽取出问题的核心实体和目标关系，在知识图谱中查询相关数据，返回查询结果。模板可以进行复用，一个模板可以覆盖多个自然语言问题。基于规则模板的问答方法可以较为准确的获得问题答案，且不涉及任何的计算推理过程，回答响应较快。但是模板建立需要大量的人工投入，且需要根据业务场景的变化不断对模板进行维护，工作量较大。

2.2.2 基于语义解析的问答方法

随着神经网络技术的成熟，基于语义解析的问答方法逐渐替代基于模板匹配的问答方法成为主流。基于语义解析的问答方法的核心思想是利用深度学习神经网络模型，让计算机理解自然语言问题的语义，将问题文本转化为模型可以理解和识别的语义表示，再将对应的语义表示转化为查询知识图谱所需的逻辑表达式，通过数据库查询引擎对逻辑表达式执行查询请求，得到最终的查询结果。模型生成的逻辑表达式通常与查询知识图谱所需的查询语言相同，例如如果用Neo4J储存知识图谱，则模型相应输出Cypher查询语句。

基于语义解析的问答方法的核心在于语义解析，其过程一般可以分为三个阶段，分别是：实体识别、关系映射和语义组合。

（1）实体识别

实体识别是自然语言处理相关技术中的一项基础技术，全称为命名实体识别（Named Entity Recognition，NER）。在自动问答方法中，实体识别的任务是将问题文本中的核心实体词抽取出来，这些核心实体词蕴含着问题文本中的有效信息，能够通过一定的映射关系转换为知识库中的特定元素（如实体、关系、属性等）。实体识别任务的核心目标是正确划分出实体描述文本的边界，以及正确分类实体描述的具体类型，为了实现这一任务目标，通常采用“序列标注”的方法来定位文本内的实体，即对文本的每一个词语进行"BIO"标签标注："B-"代表实体起始描述，后接具体的实体类型，例如"B-Company"表示该标注文本为公司实体的起始描述文本："I-"用于标注实体描述的组成文本，后接具体的实体类型，例如"I-Company"表示该单词为实体描述的组成部分，"I-"只会跟在"B-"后面或"I-"后面，不会单独出现；"O"标签表示该单词不属于任何实体。这样，一个实体描述即可由“B-”和连续的“I-”确定，实体类型则为标签所描述的类型。

实体识别任务的难点在于对实体描述边界的正确划分。对于英文文本来说，由于英文天然自带分词特点，NER任务在英文文本的识别较为容易，而在中文文本中，文本间没有明显的词语分隔符，NER 任务很大程度会受到中文分词、语法结构等因素的影响，一旦分词错误很可能导致实体边界无法正确预测。因此，多个中文 NER 方法开始转向基于字的命名实体识别研究【4】，以消除中文分词带来的语义误差。

（2）关系映射

关系映射目标是将自然语言问句中的关系短语描述映射到知识库中相应的关系或属性中。由于自然语言描述变化多样，而知识库的实体、属性的名称是固定不变的，因此需要设计一种关系映射方法，将自然语言描述的文本转化为知识库的规范描述语言，以实现数据查询表达式的构建。

关系映射的解决方法主要有两种。一种是通过预定义的字典进行关系匹配，由于数据字段数量有限，且对于限定领域问答任务来说，特定字段的描述方式是有限的（如员工人数可以描述为"职工"、"工作人数"），因此可以提前构建一个映射字典，对不同字段的不同描述情况进行覆盖，以实现自然语言到知识库字段的映射变换；另一种方式是将关系映射任务视为一个多分类任务，通过分类器对文本描述进行分类，通过神经网络等模型对自然语言描述进行语义理解，判断该描述属于哪一个知识库字段，分类判断的方法可以将自然语言语句上下文等相关语义特征一并融入计算过程，提高模型的分类能力。

（3）语义组合

语义组合是语义解析的最后一步，负责将抽取出的元素组合成对应的查询语句逻辑表达式。对于一些简单的问题，文本中只包含单独实体和简单关系，此时可以将实体直接相连构成查询语句；而对于一些复杂问题，问题描述中可能涉及多个实体以及限定条件，此时则需要考虑文本语义对多个元素进行组合，根据一定的规则生成构建查询语句。得益于深度学习相关技术的快速发展，越来越多的自动问答方法选择使用深度学习神经网络模型实现语义解析。深度学习神经网络模型拥有较强的语义理解能力和语义表达能力，2018年由Google研究团队提出BERT模型I\*2l在多项自然语言处理任务中均取得了最优的成绩。BERT模型是一种预训练模型，采用了大量的数据和深层次的模型来进行语义特征预训练，其模型参数可以被保存并使用于多种下游任务，下游任务在训练时可以从一个更丰富的语义起点出发，提高模型收敛速度与模型训练效果，降低自然语言处理任务训练成本。本文在实现多级决策的自动问答方法时，采用的IE-SQL模型即是以BERT预训练模型作为语义解析模型。

基于语义理解的自动问答方法其优点在于：如果语义解析成功，则能完整识别提问者的提问意图，从而精确地获得相匹配的查询结果。语义解析不再受限于给定的问题句式类型，使得基于语义解析的KBQA方法能够有效应对复杂多变的提问场景，具备较高的鲁棒性。但是基于语义解析的问答方法也存在一定局限：一方面，训练神经网络模型需要一定的标注数据，然而带有标注的数据集非常有限，特别是在限定领域的数据集十分稀少，很大程度上限制了模型的表现能力：另一方面，神经网络模型结构较为复杂，模型计算耗时较长，对于线上业务来说，模型计算所需的时间远大于业务所能容忍的响应时间，深度神经网络模型的实际上线应用面临一定挑战。如何平衡计算效率与计算准确性是自动问答方法需要思考的问题之一。

**第3章 证券类领域知识图谱设计**

3.1 知识采集

3.1.1 知识采集

本文通过过编写python程序下载tushare第三方包，调用tushare官网的数据下载接口，获取了基金，基金经理，基金公司，债券四类数据，并根据其类型选取了不同的属性字段。

对于基金基本数据，本文选取了的字段如下表：

| **名称** | **类型** | **默认显示** | **描述** |
| --- | --- | --- | --- |
| ts\_code | str | Y | 基金代码 |
| name | str | Y | 简称 |
| management | str | Y | 管理人 |
| custodian | str | Y | 托管人 |
| fund\_type | str | Y | 投资类型 |
| found\_date | str | Y | 成立日期 |
| due\_date | str | Y | 到期日期 |
| list\_date | str | Y | 上市时间 |
| issue\_date | str | Y | 发行日期 |
| delist\_date | str | Y | 退市日期 |
| issue\_amount | float | Y | 发行份额(亿) |
| m\_fee | float | Y | 管理费 |
| c\_fee | float | Y | 托管费 |
| duration\_year | float | Y | 存续期 |
| p\_value | float | Y | 面值 |
| min\_amount | float | Y | 起点金额(万元) |
| exp\_return | float | Y | 预期收益率 |
| benchmark | str | Y | 业绩比较基准 |
| status | str | Y | 存续状态D摘牌 I发行 L已上市 |
| invest\_type | str | Y | 投资风格 |
| type | str | Y | 基金类型 |
| trustee | str | Y | 受托人 |
| market | str | Y | E场内O场外 |

对于基金经理数据，本文选取了

| **名称** | **类型** | **默认显示** | **描述** |
| --- | --- | --- | --- |
| ts\_code | str | Y | 基金代码 |
| ann\_date | str | Y | 公告日期 |
| name | str | Y | 基金经理姓名 |
| gender | str | Y | 性别 |
| birth\_year | str | Y | 出生年份 |
| edu | str | Y | 学历 |
| nationality | str | Y | 国籍 |
| begin\_date | str | Y | 任职日期 |
| end\_date | str | Y | 离任日期 |
| resume | str | Y | 简历 |

对于基金公司数据，本文选取了

| **名称** | **类型** | **默认显示** | **描述** |
| --- | --- | --- | --- |
| name | str | Y | 基金公司名称 |
| shortname | str | Y | 简称 |
| short\_enname | str | N | 英文缩写 |
| province | str | Y | 省份 |
| city | str | Y | 城市 |
| address | str | Y | 注册地址 |
| phone | str | Y | 电话 |
| chairman | str | Y | 法人代表 |
| manager | str | Y | 总经理 |
| reg\_capital | float | Y | 注册资本 |
| setup\_date | str | Y | 成立日期 |
| main\_business | str | Y | 主要产品及业务 |
| credit\_code | str | Y | 统一社会信用代码 |

对于债券类数据，本文选取了

| **名称** | **类型** | **默认显示** | **描述** |
| --- | --- | --- | --- |
| ts\_code | str | Y | 转债代码 |
| bond\_full\_name | str | Y | 转债名称 |
| bond\_short\_name | str | Y | 转债简称 |
| cb\_code | str | Y | 转股申报代码 |
| stk\_code | str | Y | 正股代码 |
| stk\_short\_name | str | Y | 正股简称 |
| maturity | float | Y | 发行期限（年） |
| par | float | Y | 面值 |
| issue\_price | float | Y | 发行价格 |
| issue\_size | float | Y | 发行总额（元） |
| remain\_size | float | Y | 债券余额（元） |
| value\_date | str | Y | 起息日期 |
| maturity\_date | str | Y | 到期日期 |
| coupon\_rate | float | Y | 票面利率（%） |
| add\_rate | float | Y | 补偿利率（%） |
| pay\_per\_year | int | Y | 年付息次数 |
| list\_date | str | Y | 上市日期 |
| delist\_date | str | Y | 摘牌日 |
| exchange | str | Y | 上市地点 |

获取到的数据信息如图所示：

（统计图）

基金基本数据 3002 条

基金经理数据4912条

公募基金公司数据15278 条

债券类数据 1680 条

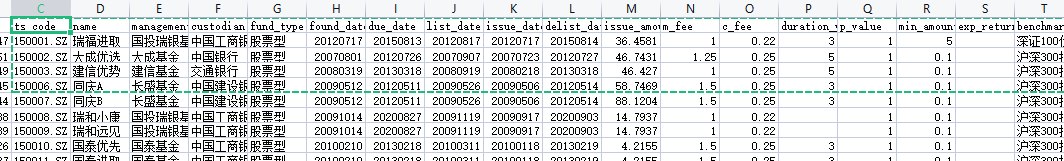
3.1.2 数据预处理

本文通过编写python程序对数据进行预处理，转换格式，删除空格，重复值，填充缺失值据。

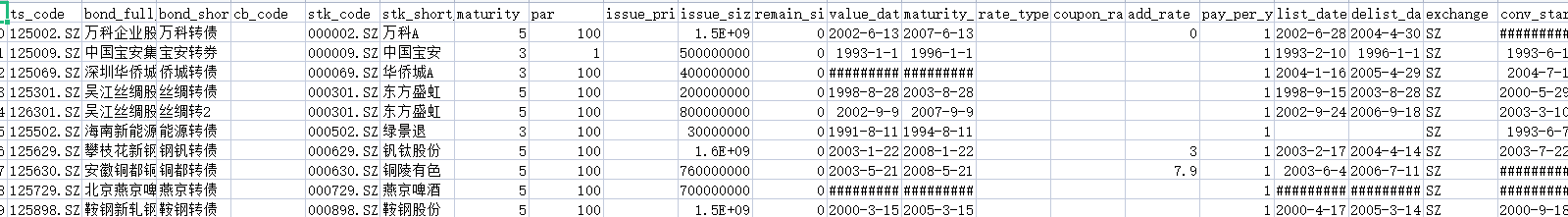
最核心的处理是将基金基本、基金经理以及基金公司三类数据进行合并，通过对数据的分析，其有共同字段ts\_code(基金代码)，根据相同的ts\_code(基金代码)将三类数据相同的一列合并为一条，并删除冗余字段。该步骤实现代码如下：

经数据清洗与合并后得到基金数据965条，证券类数据859条。

基金数据信息如下图所示：



可转债数据信息如下图所示：



3.2 知识抽取

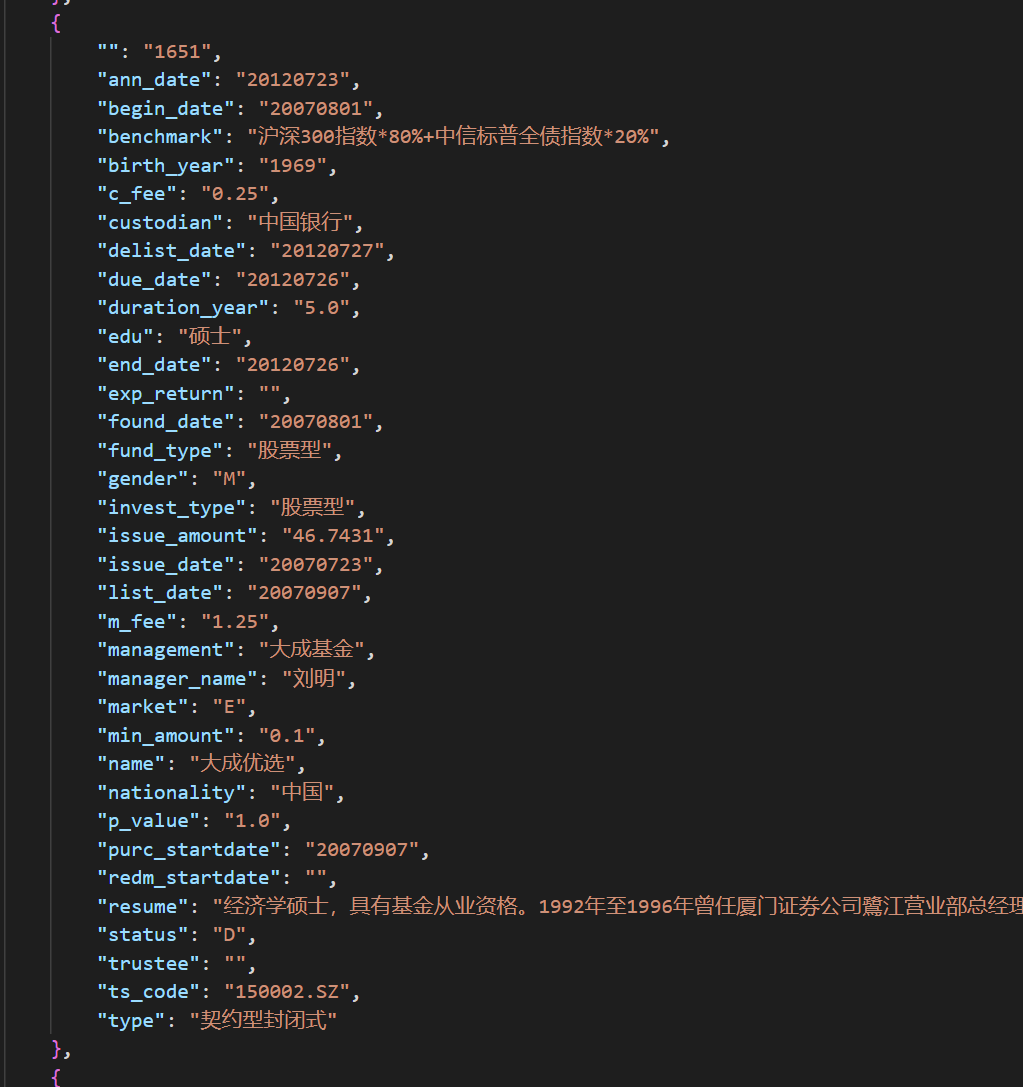
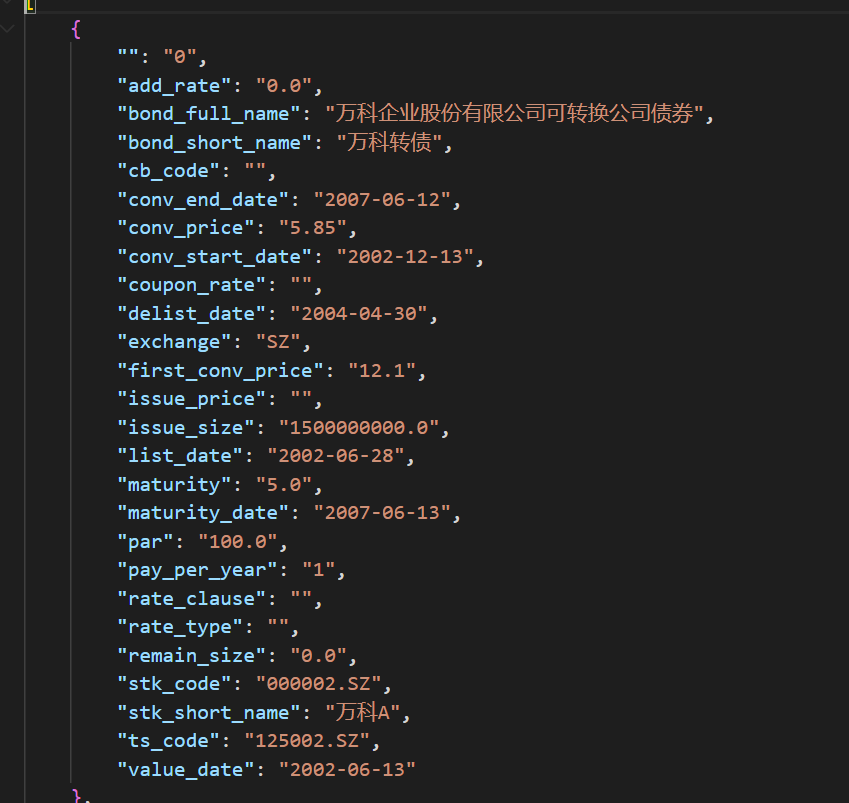
知识图谱中的数据之所以被称为知识，是因为能从其中挖掘到有用的信息。由于整个问答系统的优劣依赖于知识图谱中知识的数量与质量，因此知识抽取在构建知识图谱的过程中十分重要。

在日常生活中，我们最常见的数据是非结构化数据，所以知识抽取大多数是指从非结构化数据抽取知识的过程。对于非结构化数据进行知识抽取一般包括实体抽取、属性抽取和关系抽取等步骤。

由于半结构化数据本身比较规则，可以通过网页解析或者分析文件自身结构将其转为结构化知识数据。

本文主要选择了半结构化数据，通过编写python脚本，下载numpy第三方库，将csv文件转为json文件，其中每一个json元素对应于原有主句中的一列，键名对应原csv文件中的列名。

转换后的数据格式如下图：



其次，本文根据基金与可转债的数据格式构建了相应的规则，通过编写python程序遍历得到实体-关系-实体，实体-属性-实体，以及实体-属性这三种三元组形式，并存储在实体列表，关系列表，属性列表中。

本文根据四种不同类实体各自的属性以及之间的关系，设计了如下表：

基金：

实体类型

|  |  |  |  |
| --- | --- | --- | --- |
| 实体类型 | 中文含义 | 实体数量 | 举例 |
| ts\_code | 基金代码 |  |  |
| manager\_name | 基金经理（名字） |  |  |
| management | 管理人 |  |  |

实体关系类型

|  |  |  |  |
| --- | --- | --- | --- |
| 实体关系类型 | 中文含义 | 关系数量 | 举例 |
| manage\_by\_who | 基金经理是(被谁管理) |  | <150018.SZ,经理是,周毅> |
| belong\_to | 基金属于那个管理人 |  | <150018.SZ,属于,银华基金> |
| served\_at | 基金经理任职于 |  | <周毅,任职于,银华基金> |

基金属性

|  |  |  |
| --- | --- | --- |
| 属性类型 | 中文含义 | 举例 |
| name | 简称 |  |
| type | 基金类型 |  |
| fund\_type | 投资类型 |  |
| benchmark | 业绩比较基准 |  |
| issue\_amount | 发行份额 |  |
| m\_fee | 管理费 |  |
| min\_amount | 起点金额（万元） |  |

基金经理属性

|  |  |  |
| --- | --- | --- |
| 属性类型 | 中文含义 | 举例 |
| manager\_name | 基金经理姓名 |  |
| edu | 学历 |  |
| resume | 简历 |  |
| gender | 性别 |  |
| birth\_year | 出生年份 |  |

债券：

实体类型

|  |  |  |  |
| --- | --- | --- | --- |
| 实体类型 | 中文含义 | 实体数量 | 举例 |
| ts\_code | 转债代码 |  |  |
| stk\_code | 正股代码 |  |  |

实体关系类型

|  |  |  |  |
| --- | --- | --- | --- |
| 实体关系类型 | 中文含义 | 关系数量 | 举例 |
| match\_to | 对应正股是 |  | <125002.SZ,对应正股是,000002.SZ> |

属性类型

（以下都为转债代码的属性，正股代码属性与其重复，因此不再单独设置）

|  |  |  |
| --- | --- | --- |
| 属性类型 | 中文含义 | 举例 |
| list\_date | 上市日期 |  |
| par | 面值 |  |
| issue\_size | 发行总额（元） |  |
| bond\_full\_name | 转债名称 |  |
| bond\_short\_name | 转债简称 |  |
| pay\_per\_year | 年付息次数 |  |
| value\_date | 起息日期 |  |
| maturity\_date | 到期日期 |  |
| exchange | 上市地点 |  |

3.3 知识存储

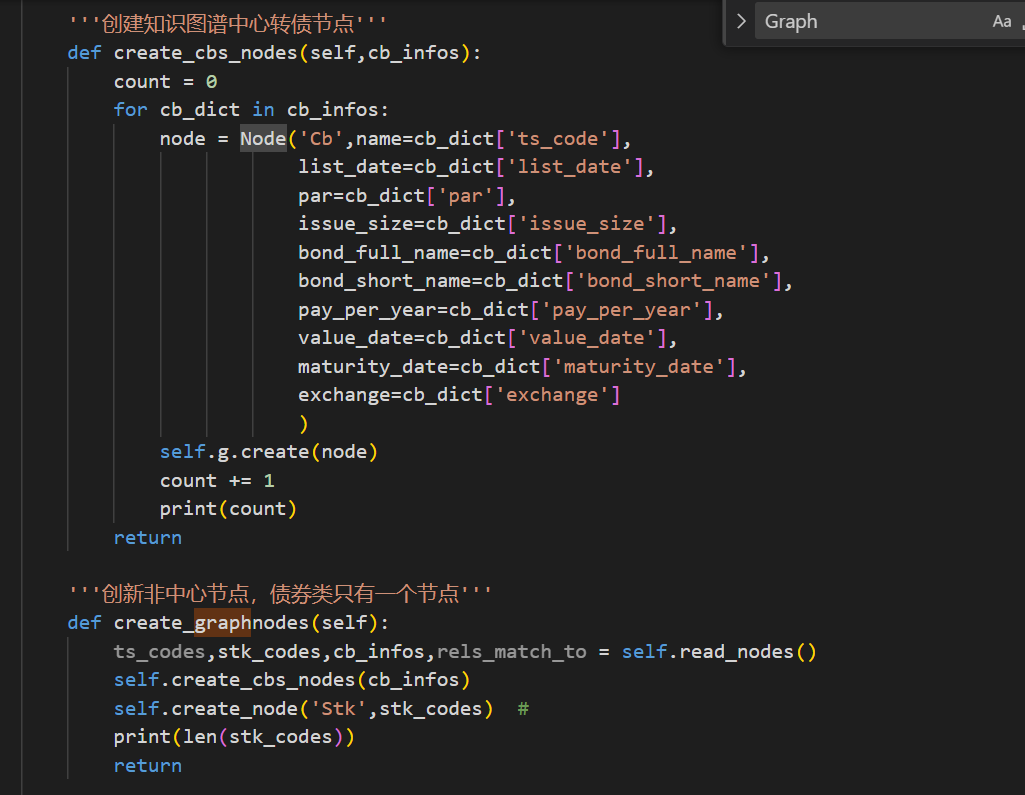
对基金和可转债数据进行知识抽取后，实现了三元组的构建，接着本文通过Python编写程序构建知识图谱的实体节点以及关系边，包含基金与可转债的证券类知识图谱就初步建立成功，最后将知识图谱存储到图数据库中。

neo4j是一个高性能的,NOSQL图形数据库，它将结构化数据存储在图和网络中，而不是表中。因为图可以更加清晰的体现各个实体间的关系，因此使本文构建的证券类知识图谱具有较强的解释性。

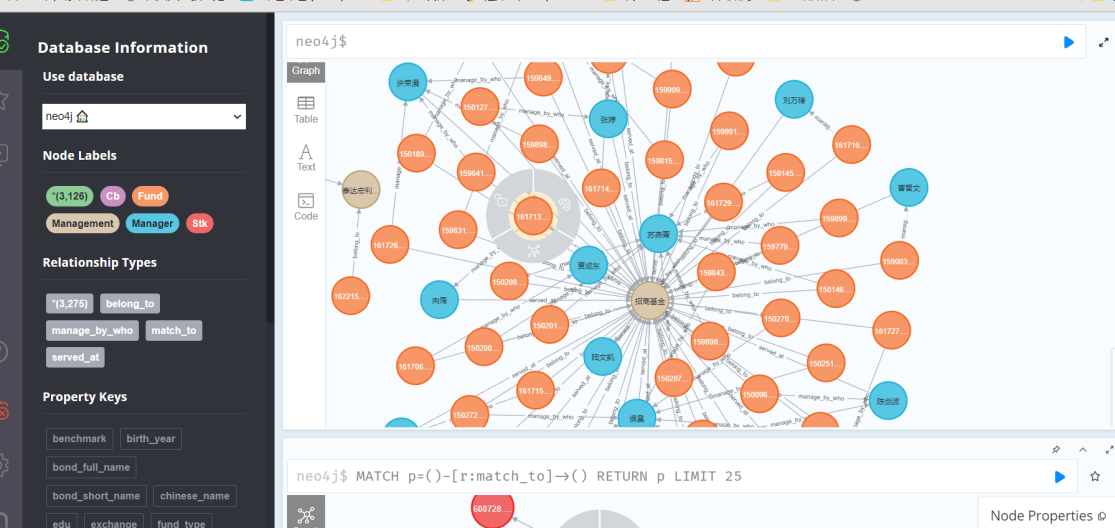
本文下载了 py2neo第三方库，通过库中的Graph（）函数连接到本地的图数据库，同时通过该库中的 Node（）方法来创建知识图谱的基金、基金经理、基金公司以及可转债债券四类实体节点以及各节点属性，再通过自定义的create\_node（）函数建立所有实体节点，create\_relationship（）函数建立所有关系，最后将实体与关系存储在图数据库中，构成了知识网络。

具体代码如下图 所示：





最后包含3126个实体，3275条关系边



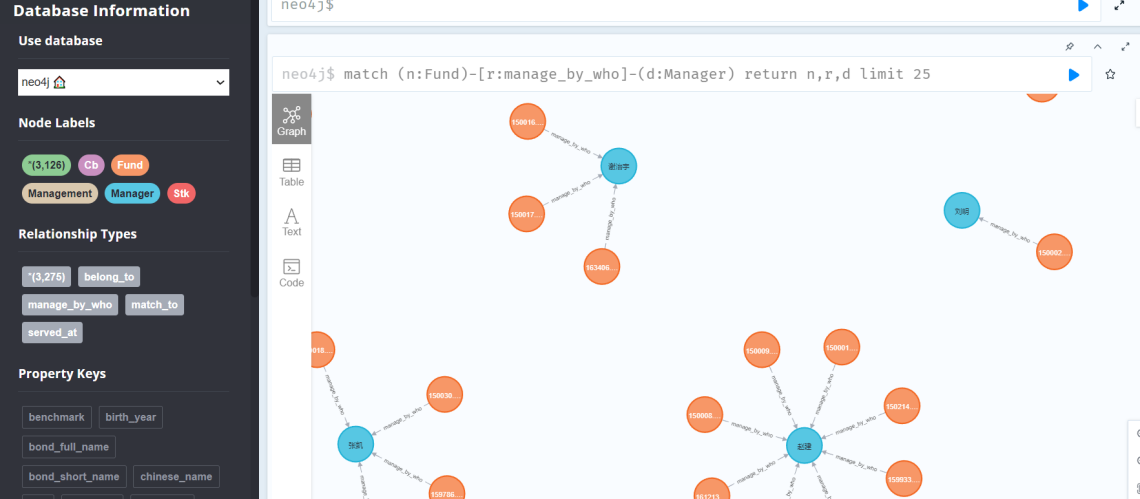
3.4 知识查询

由于[neo4j](https://so.csdn.net/so/search?q=Neo4j&spm=1001.2101.3001.7020" \t "https://blog.csdn.net/weixin_36378508/article/details/_blank)中不存在表的概念，只有标签(labels),节点(Node),关联(Relation),路径(path)，标签里存的节点,节点和关系可以简单理解为图里面的点和边,路径是用节点和关系表示的如：(a)-[r]->(b)，表示一条从节点a经关系r到节点b的路径。

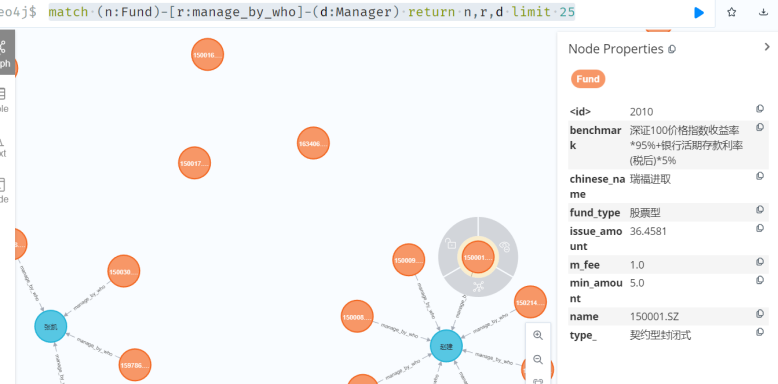
首先在后台连接上neo4j图数据库，打开相应的网址，就能看到所构建的证券类知识图谱中的可视化查询界面，在页面上输入Cypher查询语句。Cypher是一种声明式图查询语言，表达高效查询和更新图数据库，同时它具有相对简单的查询语法，事我们能够更关注业务领域问题。

例如，在neo4j中，通过Cypher查询语句“Match (n:Fund)-[r:manage\_by\_who]-(d:Manager) return n,r,d limit 25”

，查询到的结果如图3.13所示。不同颜色的节点，所代表的实体类型也不同，其中蓝色的节点代表着基金经理，橘色的节点代表着该基金经理所管理的基金，从图中可以看出谢志宇管理了三只基金，其基金代码分别为"150016.SZ"、"150017.SZ"、"163406.SZ"。



实体的属性以JSON文本形式存储，点击图中"150001.SZ"基金节点，从图3.14中可以看出基金节点"150001.SZ"的所有属性，包括中文名、基金类型、发型总额、面值等。



**第4章 基于知识图谱的问答方法实现**

4.1 基于规则的问答模型

基于规则的问答系统通常不需要复杂的算法或是深度学习等模型，它一般采用模板匹配的方式寻找匹配度最高的答案，回答结果依赖于问句类型、模板语料库的覆盖全面性，面对已知的问题，可以给出合适的答案，对于模板匹配不到的问题或问句类型，可以通过开发者自定义回答方式。

基于规则匹配的实现主要通过关键词匹配，对问句进行分类，金融领域问题本身属于封闭域类场景，本文对领域问题进行穷举并分类，然后使用cypher的match语句去图数据库中匹配查找，将查询返回的数据组装成回答语句，最后返回结果。

基于规则的证券类问答系统的流程： 首先用户输入问题，系统调用问题分类子类QuestionClassifier()对问句进行分类，若该问句没有相应的类别，则返回定义好的回答模板。反之，调用问题解析子类QuestionParser()对问句进行解析，再调用问题搜索函数子类 AnswerSearcher() 查找答案，如果有则返回结果，反之则输出定义好的回答模板。

4.1.1 定义问题模板

基于规则的问答系统所支持的问答类型完全取决于所定义的问题模板，因此本文基于金融领域中最常见的证券类问题，设计了如下问题模板：

|  |  |  |
| --- | --- | --- |
| 问句类型 | 中文含义 | 问句举例 |
| list\_date\_qwds | 可转债上市时间 | 120005.SZ多久上市 |
| par\_qwds | 可转债面值 | 120005.SZ单张面值是多少 |
| issue\_size\_qwds | 可转债发型总额 | 120005.SZ发行总额是多少 |
| bond\_full\_name\_qwds | 可转债中文全称 | 120005.SZ中文全称是 |
| bond\_short\_name\_qwds | 可转债中文简称 | 120005.SZ中文简称是 |
| pay\_per\_year\_qwds | 可转债年付息次数 | 120005.SZ一年付息几次 |
| value\_date\_qwds | 可转债起息日期 | 120005.SZ多久开始计息 |
| maturity\_date\_qwds | 可转债到息日期 | 120005.SZ什么时候到息 |
| exchange\_qwds | 可转债上市地点 | 120005.SZ在什么地方上市 |
| match\_to\_qwds | 可转债对应正股 | 120005.SZ属于哪个正股 |
| type\_qwds | 基金类型 | 501060.SH 属于什么类型 |
| fund\_type\_qwds | 基金投资类型 | 501060.SH的投资类型是啥 |
| benchmark\_qwds | 基金业绩标准 | 501060.SH的业绩参考标准 |
| issue\_amount\_qwds | 基金发行总额 | 501060.SH总共发行额是 |
| m\_fee\_qwds | 基金管理费 | 501060.SH需交多少管理费 |
| min\_amount\_qwds | 基金起点金额（万） | 501060.SH的最小投资额 |
| edu\_qwds | 基金经理学历 | 张凯的教育经历如何 |
| gender\_qwds | 基金经理性别 | 张凯是男是女 |
| resume\_qwds | 基金经理简历 | 咨询张凯的简历 |
| birth\_year\_qwds | 基金经理出生日期 | 张凯的生日是多久 |
| manage\_by\_who\_qwds | （基金）由谁（基金经理）管理 | 501060.SH的基金经理是谁 |
| served\_at\_qwds | （基金经理）任职于（基金公司） | 张凯在哪家基金公司上班 |
| belong\_to\_qwds | （基金）属于（基金公司） | 501060.SH属于哪家公司 |

4.1.2 问句分类

本文通过QuestionClassifier()类获取问句中包含的领域词及其所在领域，并收集问句当中所涉及到的实体类型；

1. 分类流程

首先调用check\_cb函数，获取问句中包含的领域词及其所在领域，并收集问句当中所涉及到的实体类型；接着基于特征词进行分类，即调用check\_word函数，看问句中是否包含某领域特征词，以及该领域是否在问句中包含的region\_words的实体类型（types）里，以此来判断问句属于哪种类型。比如：如果问句中包含基金实体，且类型为基金的起点金额，那么则将该基金起点金额的描述信息返回（question\_types = ['min\_amount\_qwds']）；若类型为该基金对应的基金经理，那么则将该基金对应的基金经理信息返回（question\_types = ['fund\_manager']）。然后将分类结果进行合并处理，组装成一个字典返回。

（2）关键实现方法：

·构建领域actree：可以加速过滤。通过python的ahocorasick库实现。ahocorasick是一种字符串匹配算法，由两种数据结构实现：trie和Aho-Corasick自动机。Trie是一个字符串索引的词典，检索相关项时时间和字符串长度成正比。AC自动机能够在一次运行中找到给定集合所有字符串。AC自动机其实就是在Trie树上实现KMP，可以完成多模式串的匹配。

·构建词典：根据四类类实体构造 {特征词：特征词对应类型} 词典

·匹配领域词：通过ahocorasick库的iter()函数匹配领域词，将有重复字符串的领域词去除短的，取最长的领域词返回。功能为过滤问句中含有的领域词，返回{问句中的领域词：词所对应的实体类型}。

4.1.3 问句解析

（1）解析流程

首先传入问句分类结果，获取问句中领域词及其实体类型。接着调用build\_entitydict函数，返回形如{'实体类型':['领域词'],...}的entity\_dict字典。然后对问句分类返回值中[‘question\_types’]的每一个question\_type，调用sql\_transfer函数转换为neo4j的Cypher语言。最后组合每种question\_type转换后的sql查询语句。

1. 关键实现方法

·解析分类结果：从上一步骤中的返回结果中获取参数，例如，从分类结果的{'args': {'张凯': ['manager']}, 'question\_types': [''manager\_resume'']}中获取args，返回{manager: ['张凯']}的形式。

·转换Cypher查询语言：不同的问题类型，转换为Cypher查询语言并返回。例如，如果问题类型为查找可转债对应的正股，则对应的Cypher语句为："MATCH (m:Cb)-[r:match\_to]->(n:Stk) where m.name = '{0}' return m.name, r.name, n.name"；

4.1.4 查询结果

（1）查询流程：

首先获取上一步骤解析的Cypher语句，调用py2neo库中的Graph.run()函数执行，得到返回结果，最后根据返回结果的类型组合答案。

（2）关键实现方法：

· 查询模块：传入问题解析的结果sqls，将保存在queries里的[‘question\_type’]和[‘sql’]分别取出，通过图数据库neo4j查询得到结果

·回复模块：根据不同问题类型，调用相应的回复模板。例如，当问题类型是 'management\_fund'，对应的回复模板是'{0} 旗下的产品有：{1}'.format(subject,'；'.join(list(set(desc))[:self.num\_limit])) 。

4.1.5 缺失实体填充

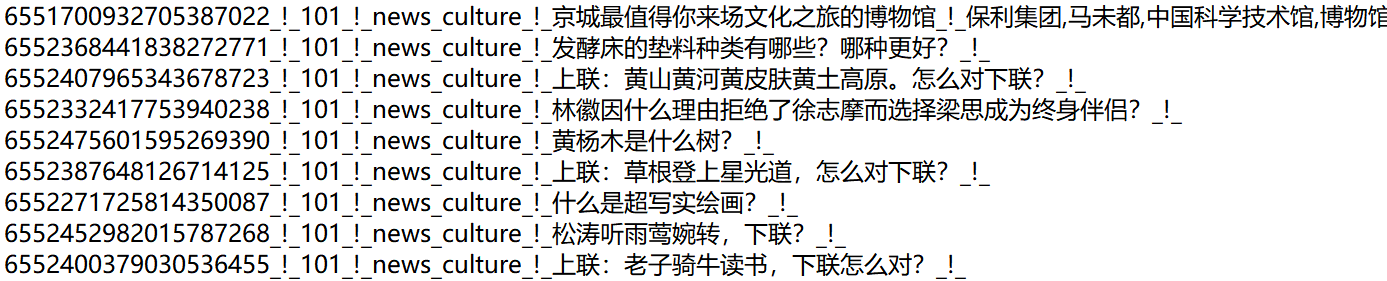
在用户连续提问时，若某轮对话中没有证券类相关的实体作为主体，默认采用上一轮对话中的主体。

4.2 基于FastText改进的问答模型

4.2.1 数据获取

本文采用数据来自今日头条数据集，共32000条，其中共有15个类别。每行为一条数据，以\_!\_分割的个字段。从前往后分别是新闻ID，分类code，类别，新闻字符串（仅含标题），新闻关键词。

数据格式如下图：



4.2.2 数据处理

在处理数据过程中，本文下载了langconv，jieba第三方库。其中langconv用于中文繁体/简体转换，jieba是一款流行并强大的中文分词工具

预处理步骤如下：

·由于数据集的TXT文档中的格式是以\_!\_来进行画划分的，根据题目条件，若要进行数据集的冗余信息的去除和保留新闻内容（关键词），就要先对\_!\_进行识别并划分

·整理停用词,将停用词txt文件转为列表

·将数据集中的繁体转为简体

·通过jieba工具数据集中的文本去除特殊符号，停用词，分词

·通过pandas库从csv文件中读取dataframe,再将数据集转为txt形式

其次，本文划分数据集为训练集与验证集，由于数据是规则排列的，直接划分势必会导致数据集的不平衡。需要先进行随机的排列生成一个新的数据，在对该数据进行划分。因此本文对于获得的数据的索引index进行随机排列，在对索引进行重新建立，根据索引对于数据进行划分按照7:3比例将数据集分割为训练集与验证集

4.2.3 模型训练

本文在准备训练模型前，首先 安装了fasttext 以及sklearn等第三方库。

首先设定模型参数，模型参数如下表：

|  |  |  |
| --- | --- | --- |
| 参数 | 含义 | 设定值 |
| dim | 词向量大小 | 100 |
| epoch |  | 100 |
| lr | 学习率 | 0.5 |
| Loss | 损失函数 | softmax |
| wordNgrams |  | 2 |
| Save\_dir | 模型保存位置 | 同项目文件夹 |

通过调用fasttext.train\_supervised（）函数进行模型训练，最后得到模型model文件。

4.2.4 模型验证

通过调用sklearn库中的classification\_report() 函数验证 fasttext模型分类效果，结果如下表所示：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| label | precision | recall | f1-score | support |
| \_label\_\_news\_world | 0.90 | 0.89 | 0.90 | 5849 |
| \_\_label\_\_stock | 0.95 | 0.93 | 0.94 | 10750 |
| \_\_label\_\_news\_car | 0.88 | 0.91 | 0.89 | 8358 |
| \_\_label\_\_news\_entertainment | 0.92 | 0.92 | 0.92 | 8092 |
| \_\_label\_\_news\_travel | 0.91 | 0.93 | 0.92 | 11771 |
| \_\_label\_\_news\_agriculture | 0.84 | 0.83 | 0.84 | 8301 |
| \_label\_\_news\_culture | 0.95 | 0.94 | 0.94 | 8784 |
| \_\_label\_\_news\_house | 0.94 | 0.93 | 0.93 | 5316 |
| \_\_label\_\_news\_story | 0.90 | 0.89 | 0.89 | 7378 |
| \_\_label\_\_news\_military | 0.97 | 0.95 | 0.96 | 11323 |
| \_\_label\_\_news\_finance | 0.86 | 0.80 | 0.83 | 1858 |
| \_\_label\_\_news\_edu | 0.88 | 0.90 | 0.89 | 12344 |
| \_\_label\_\_news\_game | 0.86 | 0.87 | 0.87 | 6494 |
| \_\_label\_\_news\_sports | 0.85 | 0.85 | 0.85 | 8074 |
| \_\_label\_\_news\_tech | 0.18 | 0.04 | 0.06 | 114 |
| accuracy |  |  | 0.90 | 114806 |
| macro avg | 0.85 | 0.84 | 0.84 | 114806 |
| weighted avg | 0.90 | 0.90 | 0.90 | 114806 |

1. **证券类问答系统的设计与实现**

本文所设计的证券类问答系统，主要包括三个部分，第一部分是证券类知识图谱的构建，利用Python编码从开源的金融类网站中获取所需的基金与债券等数据，保存在csv文件中，在对数据进行清洗与知识抽取，表示成三元组形式，并建立知识图谱的节点和边，最后将形成的知识图谱利用py2neo工具导入到neo4j图数据库中；第二部分是在问答系统的算法设计上，首先使用 jiaba分词工具对问句进行分词，并根据训练好的文本分类模型对问句分类，如果问句被分类为证券类领域，则基于规则与设置好的模板进行回答，若问句不是证券类领域问题则回答预定义的模板；最后一部分是在问答系统的web端搭建上，前端使用Vue3，Element-plus，后台使用Django框架，实现了用户登录，对话问答，提交评论，以及后台管理等功能。

5.1 需求分析

5.1.1 背景需求

随着大数据技术的发展，互联网信息以几何级数爆炸式增长，对用户快速精准的获取所需数据带来了巨大的困难。对现有的搜索引擎而言，用户通过输入框键入问题，返回结果往往不尽如人意，大部分不会给出明确答案，而是返回一些冗余的相关链接或相关程度很低的无用信息，虽然在某种程度上能够满足用户的信息搜索需求，但仍需要花费一定精力进行结果筛选，使得用户体验大打折扣。为了用户在大量杂乱无章的冗余信息中，精准快速的检索到自己想要的结果，提高用户体验度，问答系统作为搜索引擎的升级版应际而生。

当下，脱贫攻坚全面取得胜利，我国经济实力持续增长，许多家庭对金融投资、金融理财等证券类业务越发重视，将证券类知识和问答系统结合是大势所趋。基于机器学习与模板匹配的金融问答系统具有高效的问题理解能力，针对用户问题能够返回较为准确的结果，能够减少用广的检索时间，提高检索效率和用广满意度。

5.1.2 功能需求

根据背景需求并参考现有的问答系统，本文的问答系统需要实现以下功能：

1. 友好的UI界面与后端接口

前端页面为使用者提供首页入口，分别可进入问答系统与后台管理，问答系统页面中，需要为用户提供问题输入框，同时可问答页面进行布局与对话效果上的优化，引导用户输入咨询的证券类产品相关问题，并提供发送请求功能，将用户的问题文本发送给后端进行处理。管理员登录需要设置登录入口，同时在管理页面应需要有评论查看与删除等功能。

后端提供相应的接口，包括：对问题文本进行处理并返回答案，存储用户评论，校验前端管理员登录，以及管理页面查看与删除评论等接口。

（2）知识图谱问答

当用户提出问题经过问题处理后，利用分类模型与规则匹配，对问句进行分类、解析、答案搜索，首先会根据训练的分类模型对问句进行分类，若问题属于金融领域问题，再由规则匹配对问句的问题类型进行分类，并通过设定模板对问题进行解析，得到对应的cypher语句，再通过py2neo库进行知识图谱的检索，最后对答案进行拼接，完成知识图谱问答功能。

（3）非领域问题处理

当使用者的问题不是证券类领域问题，或者是证券类领域问题但问题模板无法匹配成功时，则会以闲聊和安全回答进行回复。

（4）使用反馈

用户使用问答系统后，可以对本次的使用体验进行评论，包括填写对系统的改进建议，以及描述系统在使用中出现的问题。

（5）后台管理

系统管理员可以对系统进行维护或者查看各种信息，比如查看用户填写的使用反馈，针对评论中不足的地方，对系统进行后续的改进。

5.2 运行环境

本文问答系统的运行环境如表所示

**Table 4-1 The Developing Environment Configuration of the Risk Control System**

|  |  |
| --- | --- |
| **项目** | **配置** |
| 操作系统 | Windows10 |
| 处理器 | AMD Ryzen 5 4600U |
| 开发工具 | Visual Studio Code |
| 开发语言 | Python 3.9 |
| 框架/库/包 | Django 4.1.6,jieba 0.42.1,scikit-learn 1.0.2等 |

5.2.1 开发工具

Visual Studio Code是Microsoft公司于2015年推出的一款免费的轻量级跨平台源代码编辑器，有着强大的扩展功能和生态环境，可以通过安装插件的形式大大拓展其功能如远程SSH连接、网页实时渲染、支持近二十种开发语言编辑。选用Visual Studio Code使得系统能非常便捷地连接远程服务器，直接在本机上编辑服务器上的代码并实时运行，速度快、效率高。

5.2.2 开发语言及第三方框架/库/包

（1）开发语言

后端开发语言：选用Python 3.7.12。Python是目前机器学习最常用的语言之一，它基于C语言实现。相较于C++ 等高级语言，虽然Python损失了部分性能，但它语法简单直观，容易被初学者所接纳。并且Python生态丰富，许多第三方机器学习库/框架都有Python的接口接入，例如TensorFlow、PyTorch、keras等。因为联邦学习中需要使用第三方框架来进行客户端与中心服务器的数据交互，所以选用Python作为开发语言。

前端开发语言：选用Vue3。Vue.js是一套用于构建用户界面的渐进式框架。与其它大型框架不同的是，Vue 被设计为可以自底向上逐层应用。Vue 的核心库只关注视图层，不仅易于上手，还便于与第三方库或既有项目整合。另一方面，当与现代化的工具链以及各种支持类库结合使用时，Vue 也完全能够为复杂的单页应用提供驱动。

数据库：选用Neo4j。Neo4j是一款非关系型图数据库，其存储结构和有向图相似，由结点和与节点连接的关系为基本存储内容，由于没有传统的表结构，其最大的特点是扩展性好，适合存储半结构化数据，可以为用户提供关系数据存储。被广泛应用于社交网络、智能推荐引擎、知识图谱等。能够为用户提供数据处理、数据可视化、知识检索等操作。

（2）第三方框架/库/包

·JieB：是一款强大的中文分词工具，结合了基于规则和基于统计这两类方法，同时支持支持三种分词模式：精确模式，全模式，搜索引擎模式，支持繁体分词，支持自定义词典，MIT 授权协议

·Django：Django是一款开源代码框架，它由Python编写，用于开发web应用，编程人员只需要开发少量的代码，就可以实现后台管理。Django采用MTV设计模式，其发展于传统的MVC设计模式，为了避免开发过程中引入过于复杂的SQL语句，它把数据库表抽象成Python类，将用户对数据库表的操作转化为对Python 类的操作。

· Pandas：Pandas是python的一个[数据分析](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%88%86%E6%9E%90/6577123?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/pandas/_blank)包，最初由AQR Capital Management于2008年4月开发，并于2009年底开源出来，目前由专注于[Python](https://baike.baidu.com/item/Python/407313?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/pandas/_blank)数据包开发的PyData开发团队继续开发和维护，属于PyData项目的一部分。Pandas最初[被作为](https://baike.baidu.com/item/%E8%A2%AB%E4%BD%9C%E4%B8%BA/6051803?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/pandas/_blank)金融数据分析工具而开发出来，因此，pandas为[时间序列分析](https://baike.baidu.com/item/%E6%97%B6%E9%97%B4%E5%BA%8F%E5%88%97%E5%88%86%E6%9E%90/8724605?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/pandas/_blank)提供了很好的支持。 Pandas的名称来自于[面板数据](https://baike.baidu.com/item/%E9%9D%A2%E6%9D%BF%E6%95%B0%E6%8D%AE/6483369?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/pandas/_blank)（panel data）和python数据分析（data analysis）。panel data是经济学中关于多维数据集的一个术语，在Pandas中也提供了panel的[数据类型](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E7%B1%BB%E5%9E%8B/10997964?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/pandas/_blank)。

·scikit-learn：Scikit-learn主要是用[Python](https://baike.baidu.com/item/Python/407313?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/_blank)编写的，并且广泛使用[numpy](https://baike.baidu.com/item/numpy/5678437?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/_blank)进行高性能的[线性代数](https://baike.baidu.com/item/%E7%BA%BF%E6%80%A7%E4%BB%A3%E6%95%B0/800?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/_blank)和数组运算。此外，用Cython编写了一些核心算法来提高性能。[支持向量机](https://baike.baidu.com/item/%E6%94%AF%E6%8C%81%E5%90%91%E9%87%8F%E6%9C%BA/9683835?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/_blank)由围绕[LIBSVM](https://baike.baidu.com/item/LIBSVM/10483771?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/_blank)的Cython[包装器](https://baike.baidu.com/item/%E5%8C%85%E8%A3%85%E5%99%A8/678046?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/_blank)实现；逻辑回归和线性支持向量机的相似包装围绕LIBLINEAR。在这种情况下，可能无法使用Python扩展这些方法。Scikit-learn与许多其他Python库很好地集成在一起，例如matplotlib和plotly用于[绘图](https://baike.baidu.com/item/%E7%BB%98%E5%9B%BE/1274830?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/_blank)，numpy用于数组矢量化，[pandas](https://baike.baidu.com/item/pandas/17209606?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/_blank)[数据帧](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%B8%A7/10571824?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/_blank)，scipy等。

5.3 系统实现

5.4.1 系统总体设计

（ 框架图）

5.4.2 逻辑层实现

逻辑层是整个问答系统的关键之处，这一模块主要包括问答流程的所有算法，为前端展示模块提供对外服务API接口。首先根据今日头条的文本数据进行训练，保存文本分类模型。接着对用户提出的问句进行专业分类，得到关键信息和查询模板，映射到知识图谱中。最后在知识图谱中进行查询，将答案返回给用户。

5.4.3 数据层实现

数据模块为证券类问答系统提供数据支撑，主要包括存储在neo4j图数据库中的知识图谱，用于答案查询；利用知识图谱中的数据生成问句语料库，用于模板匹配；利用今日头条数据进行模型训练，用于文本分类。

5.4.4 前端展示层实现

前端展示模块主要用于实现系统和用户的交互，系统界面如图5.2所示。系统问答过程如下：用户首先打开系统页面，然后在对话框中输入问题并点击"答案查询"按钮提交问答申请，通过Get方式将申请提交给后端；后端先获取到用户所提问题，再将问题反馈给问答模块，从而得出答案；最后将得到的答案返回给前端展示模块。

5.4 系统演示和测试

5.5.1 演示

5.5.2 测试

**第6章 总结与展望**

6.1 课题工作总结

近年来，人们越来越多地购买金融证券类产品，人们常通过引擎搜索或者向专业机构人员咨询来获得信息。但金融机构人员每天面对大量用户，精力有限，无法第一时间就给用户提供服务。而通过搜索引擎查询的话，则会返回一系列网页链接，挑选出正确信息需要用户耗费大量的时间。并且搜索引擎仅考虑问句与答案在字级别上的相似度，难以理解用户的真实想法，有时返回的答案与想要信息是毫不相关。问答系统在使用者提问后、能够精准明白用户真实目的、直接返回精炼过的答案。同时，知识图谱又为问答系统提供了坚实的数据支撑。因此，本文将金融证券类信息与自然语言处理技术结合，设计并实现了一个基于知识图谱的证券类问答系统。本文工作内容如下：

1. 金融知识图谱构建。构建过程包括知识采集、知识抽取、知识存储。首先确定数据源，数据质量的好坏会影响问答的效果，本文最终选择了Tushare大数据社以及其他一些金融网站作为数据来源。Tushare大数据社区的数据可以通过接口调用，直接下载。数据获取完并存储在本地后，还需对数据进行清洗。根据制定的规则对获取的数据进行实体，属性和关系的抽取，采用自底向上的方式构建证券类领域的知识图谱，最后将内容存储到neo4i图数据中。
2. 基于知识图谱的问答关键技术研究。核心是将用户所提问题中的关键字链接到知识图谱中，返回直观简洁的答案。这一过程主要包括两个部分：问句解析和答案检索。本文首先在构建的知识图谱的基础上，根据常见的用户问题，构建查询模板，其次再对今日头条的数据进行训练，使其能够对用户的问题进行准确分类，当问题被分为金融领域问题时，再通过设定好的模板进行匹配，若分类为其他领域或者匹配失败则返回闲聊语句，最后对查询结果，对关键信息进行填充，转化为cypher语句，完成知识图谱的查询和推理，实现金融智能问答。
3. 金融问答系统搭建。使用Django框架搭建基于Web端的金融问答系统，整个系统分为三个模块，数据模块、逻辑模块、前端展示模块。数据模块为金融问答系统提供数据支撑，逻辑模块主要包括问答流程的相关算法并为前端展示模块提供对外服务API接口，前端展示模块主要用于实现系统和用户的交互。

6.2 未来工作展望

在面向证券类领域的自动问答系统的研究与实现中，仍存在一些问题和工作待进一步的深入研究。整个问答系统的优劣依赖于知识图谱中知识的数量与质量。也算是利弊共存吧！知识图谱图谱具有良好的可扩展性，扩展了知识图谱也就是扩展了问答系统的知识库。如果问句在射程范围内，可轻松回答，但如果不幸脱靶，则体验大打折扣。未来进一步工作可以从以下方面着于开展：

1. 知识图谱中的知识来源太单一，都是从半结构化数据中获取，未来可以从非结构化数据中抽取信息，进行知识融合和加工，丰富图谱的多样性。
2. 本文的系统主要基于常见的问题设定模板，使用问句匹配的方式实现的，未来可以使用BiLSTM-CRF神经网络模型进行实体识别。本文问题分类采用的是机器学习模型，未来可以尝试CNN等神经网络模型。
3. 本系统针对的都是简单的单轮对话，未来希望能拓展复杂的多轮对话知识图谱中的知识来源太单一，都是从半结构化数据中获取，未来可以从非结构化数据中抽取信息，进行知识融合和加工，丰富图谱的多样性。

**参考文献**

**声明**

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。据我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得四川大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

本学位论文成果是本人在四川大学读书期间在导师指导下取得的，论文成果归四川大学所有，特此声明。

学位论文作者（签名）\_\_\_\_\_\_\_\_\_\_

论文指导教师（签名）\_\_\_\_\_\_\_\_\_\_

2023年5月20日

**致谢**

时光飞逝，四年的本科生学习生活转眼即逝，我非常庆幸能够进入四川大学计算机学院，在这样一个大家庭中学习、生活与工作感觉到无比的幸福。回首往日时光，任课老师的谆谆教导、同学们的互帮互助不仅让自己收获了知识，提升了专业技能，更增强了我的综合能力，这段经历必将成为我人生道路上浓墨重彩的一笔。

感谢我的毕业论文指导老师段磊教授在科研、学习上给予我的关怀和指导。这篇论文从选题、开题、研究以及实现到毕业论文的撰写都离不开段老师的悉心指导。段老师提供了有价值的意见，使我顺利完成了论文的研究和撰写工作。段老师治学态度严谨，具有深邃的洞察力和前瞻性的眼光，不仅在论文研究过程中给予了我宝贵意见和指导，在对待科研与工作的态度上也潜移默化的影响了我。段老师丰硕的学术成果及经历为我明确了未来的研究方向和奋斗目标。

感谢我的研究生导师章毅教授在我保研后的学习、科研、生活上给予全力的支持。在我迷茫的时期，章老师给予我了深切的关心和鼓励，在科研何生活上指明了方向，对此我深表感激

感谢辅导员马力老师在我本科学习生活上的帮助与鼓励，在马老师的温馨陪伴下，让我能够健康快乐充实地度过这四年本科生活。

衷心感谢我的父母，你们的养育之恩我无以为报。我的种种品质均源于他们的悉心教育，他们的关怀使我在人生道路上更加勇敢地挑战苦难并战胜困难。

最后，感谢各位评审老师在百忙之中抽出时间评审这篇论文并提出宝贵的意见与指导。我要向我的父母表示感谢，你们细致的关心与无条件的支持是我坚实的后盾，支持我一路走来直至本科毕业。我还要向我的母校四川大学表示感谢，我即将继续在四川大学开启又一个三年，川大人的标签已深深烙印在我的心里。

漫漫人生路，感恩引路人。文有所长，不胜感激。纵有不舍，终有别离。愿下次相见，仍美好如初。山河无恙，未来可期。