分 类 号 学号

学校代码 10487 密级



硕士学位论文

基于循环分段残差网络的对抗关系抽取研究与实现

学位申请人：赵恢强

学科专业：信息与通信工程

指导教师：王玉明 副教授

答辩日期：2019年5月25日

A Dissertation Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Engineering

Design and Implementation of Adversarial Relation Extraction Based on Recurrent Piecewise Residual Networks

Candidate： Huiqiang Zhao

Major： Information and Communication Engineering

Supervisor：Prof. Wang Yuming

**Huazhong University of Science and Technology**

**Wuhan, Hubei 430074, P. R. China**

**May, 2019**

**独创性声明**

本人声明所呈交的学位论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除文中已经标明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的研究成果。对本文的研究做出贡献的个人和集体，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：

日期： 年 月 日

**学位论文版权使用授权书**

本学位论文作者完全了解学校有关保留、使用学位论文的规定，即：学校有权保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权华中科技大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

保 密□，在\_\_\_\_\_年解密后适用本授权书。

不保密□。

（请在以上方框内打“√”）

学位论文作者签名： 指导教师签名：

日期： 年 月 日 日期： 年 月 日

摘 要

作为信息抽取的核心任务，有效的关系抽取是信息爆炸时代的一大挑战。本文在进行关系抽取这一基础性研究时发现，现有的模型都无法有效的抽取长而复杂或无直接关系触发词的句子中的实体对关系，而这种句子在自然语言文本中是普遍存在的，这使得关系抽取的效果遇到了瓶颈。

本文深入调研并分析了国内外在关系抽取方面的研究。在融合更多更深次信息的思想启发下，为了丰富模型的表示，提出**实体类型嵌入(Entity Type Embedding, ETE)**，ETE和词嵌入、位置嵌入一起作为模型的联合嵌入层；为了更精细地编码模型的嵌入表示，精心设计了**循环分段残差网络(Recurrent Piecewise Residual Networks, RPRN)**结构，RPRN能有效获取句子上下文中的潜在表示。本文在进行关系抽取建模时还引入了对抗训练，从而进一步增强模型的泛化能力。本文提出的模型有效解决了现存模型所遇到的困境，并在纽约时报公开数据集上进行了验证，实验结果显示本文提出的胜过现存最先进的模型。

关系抽取不仅具有很高的理论研究价值，而且具有非常广泛的实际应用。本文将提出的模型应用到了科研文档中，并应用知识图谱可视化技术构建了一个语义检索系统。通过机器学习相关技术自动抽取自然语言文本语料中蕴含的关系事实，人们可以更加有效地获取文本中涉及的各主客体之间的语义关系，避免了以人工阅读的方式从不断生成的海量文本中提取关系事实等结构信息，这对文本语料的信息化建设具有重大的意义。同时抽取的关系事实是构建语义检索、问答系统等应用的重要基础。

**关键词：**关系抽取；实体类型嵌入；循环神经网络；残差网络

Abstract

As a core mission in information extraction, effective relation extraction is a great challenge in the information explosion era. In this paper, we found that existing models usually fail in extracting correct relation of entity pair effectively when the sentence is long and complex or there is not the firsthand trigger word of corresponding relation, whereas such sentences are ubiquitous in texts formed of natural language, which makes the effects of relation extraction hit a bottleneck.

In this paper, we penetrate into investigating and anlyzing the researches on relation extraction at home and abroad. Inspired by the idea of fusing more and deeper information, we propose **Entity Type Embedding(ETE)** to enrich the representation of the model, ETE, word embedding and position embedding serve as the joint embedding of our model; For encoding the representation of the embedding subtly, we devise an architecture of **Recurrent Piecewise Residual Networks(RPRN)** meticulously, which may acquire the latent representation underlying the context of a sentence. We introduce adversrial training when modeling relation extraction as well, which further enhance generalization capacity of our model. Our proposal effectively solve the dilemma of existing models, and is validated by experiments on public dataset of New York Times. Experiment results show that our proposal outperforms the existing state-of-the-art models.

Relation extraction is not only of great values for theory study, but also has extremely extensive application. We apply proposed model to documents of scientific research, and employ visuallization technology of knowledge graph to build a system of semantic search. Extracting the relation facts implied in the texts formed of natural language automatically by the technology of machine learning, pepole are able to acquire semantic relations between the subjects and objects involved in the text more effectively, and get rid of extracting structural information such as relation facts from the massive texts constantly generated by means of artificial reading, which is of great significance to information construction of text corpus. Simutaneously, the extracted relation facts are key foundation of building applications such as semantic search, question answering system and etc.

**Keywords**：Relation Extraction, Entity Type Embedding, Recurrent Neural Networks, Residual Networks

# 目 录

[摘 要 I](#_Toc1721383)

[Abstract II](#_Toc1721384)

[目 录 III](#_Toc1721385)

[略缩语 V](#_Toc1721386)

[1 绪论 1](#_Toc1721387)

[1.1 课题背景及来源 1](#_Toc1721388)

[1.2 课题现状及分析 2](#_Toc1721389)

[1.3 研究内容及意义 5](#_Toc1721390)

[1.4 本文的组织结构 6](#_Toc1721391)

[2 相关理论及技术概述 7](#_Toc1721392)

[2.1 图论介绍 7](#_Toc1721393)

[2.2 SSH框架 13](#_Toc1721394)

[2.3 面向服务架构 14](#_Toc1721395)

[2.4 数据可视化 15](#_Toc1721396)

[2.5 本章小结 16](#_Toc1721397)

[3 关系抽取分析与设计 17](#_Toc1721398)

[3.1 总体需求分析 17](#_Toc1721399)

[3.2 总体架构设计 18](#_Toc1721400)

[3.3 数据清洗模块 21](#_Toc1721401)

[3.4 协作网络建模 25](#_Toc1721402)

[3.5 协作推荐模型 29](#_Toc1721403)

[3.6 本章小结 37](#_Toc1721404)

[4 关系抽取实现 38](#_Toc1721405)

[4.1 开发环境搭建 38](#_Toc1721406)

[4.2 数据清洗模块实现 39](#_Toc1721407)

[4.3 协作网络建模实现 46](#_Toc1721408)

[4.4 协作推荐模型实现 50](#_Toc1721409)

[4.5 本章小结 60](#_Toc1721410)

[5 模型评估及案例分析 61](#_Toc1721411)

[5.1 系统测试及性能分析 61](#_Toc1721412)

[5.2 协作关系网络探究 66](#_Toc1721413)

[5.3 本章小结 72](#_Toc1721414)

[6 总结与展望 73](#_Toc1721415)

[6.1 论文总结 73](#_Toc1721416)

[6.2 论文展望 74](#_Toc1721417)

[致 谢 75](#_Toc1721418)

[参考文献 76](#_Toc1721419)

# 略缩语

|  |  |
| --- | --- |
| **ETE** – Entity Type Embedding | 实体类型嵌入 |
| **RPRN** – Recurrent Piecewise Residual Networks | 循环分段残差网络 |
| **NYT** – New York Times | 纽约时报 |
| **CNN** – Convolutional Neural Networks | 卷积神经网络 |
| **RNN** – Recurrent Neural Networks | 循环神经网络 |
| **LSTM –** Long Short Term Memory | 长短时记忆 |
| **GRU** – Gated Recurrent Unit | 门控循环单元 |
| **NLP** – Natural Language Processing | 自然语言处理 |
| **POS** – Part Of Speech tagging | 词性标注 |
| **SRL** – Sematic Role Labeling | 语义角色标注 |
| **NER –** Named Entity Recognition | 命名实体识别 |
| **DARPA -** Defense Advanced Research Projects Agency | 国防高级研究计划局 |
| **NIST -** National Institute of Standards and Technology | 国家标准与技术研究所 |
| **MUC –** Message Understanding Conference | 消息理解会议 |
| **ACE –** Automatic Content Extracting | 自动内容抽取 |
| **OCR –** Optical Character Recognition | 光学字符识别 |
| **CRF -** Conditional Random Field | 条件随机场 |
| **MIML –** Multiple Instance Multiple Label | 多实例多标签 |
| **KB –** Knowledge Base | 知识库 |
| **BP** –Backpropagation | 反向传播 |
| **BPTT –**Backpropagation Through Time | 通过时间反向传播 |
| **ResNet –** Residual Networks | 残差网络 |
| **VGGNet** –Visual Geometry Group Network | 视觉几何群网络 |
| **ILSVRC** – Imagenet Large Scale Visual Recognition Challenge | 大规模视觉识别挑战赛 |
| **COCO –** Common Objects in Context | 上下文公共对象 |
| **ReLU –** Rectified Linear Units | 整流线性单元(激活函数) |
| **CPUs –** Central Processing Units | 中央处理器 |
| **GPUs –** Graphics Processing Units | 图形处理器 |
| **TPUs –** Tensor Processing Units | 张量处理器 |
| **ASICs** **–** Application Specific Integrated Circuits | 专用集成电路 |

# 绪论

随着信息科技的不断发展，人们的工作和生活变得越来越便捷。然而面对信息爆炸的数据洪流，要想从中获得有用信息则变得举步维艰。世界上的许多知识是用自然语言文字记录的，但如何以这种形式有效地加以利用是一项重大挑战[1]。本课题研究的关系抽取正是信息抽取[1][2]的一项核心而又艰巨的任务，同时又是构建语义搜索[3]、问答系统[4]等知识图谱[5]相关应用的重要基础。

近年来，关系抽取的研究越来越受到国内外研究学者的关注，在这方面的研究也取得了显著的进展。但是由于关系抽取这个任务本身的复杂性，现有的模型依然存在许多的局限，关系抽取的效果差强人意。

本课题就是要研究并解决目前的关系抽取模型所面临的困境，如无法有效的抽取长而复杂或无直接关系触发词的句子中涉及的实体对关系。在本章中，首先，简要介绍了本课题的背景及来源；其次，概述了国内外在关系抽取方面的研究，并剖析了关系抽取当前所面临的困境及本课题要解决的重难点问题；然后，简明扼要的阐述了本课题研究的主要内容是什么，以及研究的意义在哪里；最后，说明了下本文的行文组织结构。

## 课题背景及来源

由于网络应用爆炸式增长，互联网中每时每刻都有海量的数据在不断地生成，这标志着大数据、云计算的时代已经来临。这一方面给人们的生活方式带来了极大的变革，另一方面，却也使得人们从这些海量数据中获取有效信息变得愈发困难。因此我们亟需一种能够从海量的自然语言文本中抽取有用信息的技术，从而更加充分发挥这些文本语料的价值，信息抽取正是应上述之需而诞生的。关系抽取是信息抽取中极为重要而又复杂的任务，然而当前已有的模型在关系抽取上的效果却不尽人意。因此，关系抽取的研究依然任重而道远。

本课题来源于教育部社科研究重大委托课题“科研管理信息化建设”，并受到教育部哲学社科繁荣计划建设项目（2017-2018）、华中科技大学双一流建设项目（2016-2018）、华中科技大学自主创新研究基金项目（No.2015AE027）的资助。本课题针对当前已有模型遇到的困境，以丰富模型的嵌入层表示及编码模型的潜在表示为切入点，提出新的嵌入模型，并设计更好的神经网络结构，来建模关系抽取。同时将本文提出的模型应用到社会科学研究文档中，并构建一个语义检索系统。构建的系统用于为高校社科管理人员提供服务，进而推进科研管理的信息化建设。

## 课题现状及分析

### 国内外研究概况

由美国国防高级研究计划局(DARPA)资助的最后一届消息理解会议(**MUC**)首次引入关系抽取任务[6]。美国国家标准技术研究院(NIST)组织的自动内容抽取(ACE)评测会议中提出的一项重要任务就是实体关系识别[7]。ACE评测会议中关系语料的来源包括新闻专线、新闻广播、通过OCR识别的报纸、新闻组、电视访谈、网络日志等等。与MUC相比，ACE的数据规模和语种数量都有了大幅度提升。MUC、ACE评测会议的实体关系抽取涉及的关系类型局限于命名实体( 包括人名、地名、组织、机构名等)之间的少数几种类型的实体关系，如雇佣关系、地理位置关系、人—社会组织关系等。SemEval[8]是继MUC、ACE后信息抽取领域又一重要评测会议，该会议吸引了大量的院校和研究机构参与测评。SemEval-2007[9]定义了普通名词或名词性短语之间的语义关系，但提供的语料库规模较小，SemEval-2010[10]任务8在SemEval-2007基础上进行了丰富和完善，引发了普通名词或名词性短语之间语义关系抽取的研究新高潮。

关系抽取的研究越来越受到国内外研究学者的关注，研究的方法也是层出不穷，概括起来主要包含基于传统有监督的方法，基于远程监督的方法，以及基于深度学习的学习方法三大类。

1. 基于传统有监督的方法

在关系抽取中，目前有监督的方法能抽取更有效的特征，其准确率和召回率都更高，因此有监督的方法受到了国内外研究学者更多的关注。

有监督的关系抽取方法需要标注好的训练语料，从标注语料中自动学习关系对应的抽取模式。典型的方法有：Miller, Scott等人[11]的基于句法解析增强的方法, Kambhatla, Nanda等人[12]的基于逻辑回归的方法,基于核的方法[13],包括Zhao and Grishman等人[14]的解析树核、Bunescu and Mooney等人[15]的子序列核等, Culotta, Aron等人[16]的基于条件随机场(CRF)的方法。然而在关系抽取中,阻碍这些方法的主要问题在于人工标注训练语料耗时耗力。

1. 基于远程监督的方法

针对传统有监督方法存在的上述局限性，Mintz et al[17],Min et al [18],Angeli et al [19]等人将远程监督的思想应用到了关系抽取的建模过程中。他们将纽约时报(NYT)新闻语料与大规模知识库FreeBase[20](包含7300多个关系和超过9亿个实体)进行实体对齐。远程监督假设为：一个包含两个实体的句子蕴含了该实体对在知识库当中的语义关系，然后将这个句子作为该实体对在知识库中所蕴含关系的训练正例。作者使用远程监督方法标注训练语料，然后抽取语料特征并训练关系抽取学习器，有效解决了关系抽取的标注训练语料规模的问题。后续众多研究学者从不同角度对远程监督技术进行了改进。例如，Takamatsu等人[21]提出了实体对齐技术的改进方案，缓解了噪声数据带来的性能下降问题；Yao等人[22]提出了基于无向图模型的实体关系抽取方法；Riedel等人[23]则增强了远程监督的假设，与 Mintz的相比错误率减少了31%，提高了关系抽取的整体效果。

上述基于远程监督的假设均是一个实体对只对应一种关系。然而实际中许多实体对具有多重关系。如“Steve Jobs founded Apple”和“Steve Jobs is the CEO of Apple”就表达了实体对之间的多重关系。为解决在很多实际场景中基于上述假设的不合理性，Hoffmann等人[24]提出了多实例多标签(MIML)方法来建模关系抽取，考虑了一个实体对存在多重关系的场景。类似地，Surdeanu等人[25]也提出了多实例多标签+贝叶斯网络(MIML+BN)来抽取自然语言文本语料中实体之间的语义关系。关于生成负样本的这一严重缺陷，Bonan等人[26]提出了在实体对级别上仅仅从正例与未标记样例中学习的算法，Angeli等人[27]提出了一种利用少量精心选取的实例对远程监督关系抽取器进行局部监督的方法。

1. 基于深度学习的方法

应用传统机器学习的方法尽管已经取得了很不错的效果，但这些方法过于依赖词性标注(POS)[28]、语义角色标注(SRL)[29]等自然语言处理(NLP)[30]任务。这些任务在经过NLP工具处理后存在大量的错误，而这些错误在关系抽取中被不断传播放大，进而影响关系抽取的整体性能。近年来，由于深度学习的迅速崛起，尤其是其在计算机视觉领域的革命性突破，越来越多的研究学者倾向于用深度学习的方法来建模关系抽取。值得关注的一点是，深度学习允许由多个处理层组成的计算模型学习具有多个抽象级别的数据表示[31]。

Socher等人[32]提出采用递归神经网络来建模实体关系抽取，该方法考虑了句子的句法结构信息，但没有考虑两个实体在句子中的位置及语义信息。Zeng et al[33], Nguyen et al[34],Xu et al[35]等人将卷积神经网络(CNN)运用于实体关系抽取任务中，该方法同时考虑实例的词向量和词的位置向量，在关系抽取中较好地考虑到了句子中的实体信息。之后，Santos等人[36]提出了一种新的卷积神经网络应用于实体关系抽取，并使用了新的损失函数，更加有效地区分了不同的关系类别。

深度学习方法在人工标注的数据集上取得了巨大成功，但是与基于特征的关系抽取方法类似，基于深度学习方法的关系抽取模型也面临着人工标注语料耗时耗力的问题。为此，Zeng等人[37]提出通过CNN与远程监督结合来进行关系抽取，该方法在一定程度上解决了神经网络关系抽取模型在远程监督数据上的应用，在NYT数据集上取得了远远高于基于特征的关系抽取模型的预测效果。Miwa等人[38]提出端到端的神经网络，该网络用双向LSTM[39]来建模关系抽取。Peng et al[40]、Song et al[41]等人探究了一种基于图LSTM的通用关系抽取框架，该框架很容易扩展到跨句子的n元关系抽取。为了有效缓解噪声数据带来的性能下降问题，Lin等人[42]将选择性注意力机制应用到了关系抽取中，该方法可以根据指定关系为袋子中的每个实例分配权重，通过训练学习从而使有效实例的权重不断提高，而带噪声的实例权重不断下降。在其他相同情况下，与不带注意力机制的模型相比，该方法的整体效果提升较大。类似地，Wang et al[43]、Du et al[44]等人提出一种新的多级别的注意力机制用于关系抽取。Wu et al[45]、Qin et al[46]等人将对抗模型引入到关系抽取中，从而进一步增强了模型的泛化能力。此外，也有改善子句模式的自训练[47]、全局关系嵌入[48]等各种其他的关系抽取方法。

### 重点难点问题

本课题对关系抽取这一基础性任务进行了研究，为提高关系抽取的效率，提出和设计了新的关系抽取模型，并将提出的模型应用到科研文档中，以构建一个语义检索系统。本课题在进行关系抽取的研究与实现过程中，重点问题主要为以下几点：

1. 大规模语料的标注

语料标注在关系抽取建模过程中是一个非常重要的问题，尤其是对于大规模语料，人工标注耗时耗力，这就亟需一种自动标注语料的技术。本课题是在关系抽取研究中广泛采用的NYT数据集上进行实验的，该数据集规模庞大，本文采用了远程监督[17]技术来解决语料标注的问题。

1. 噪声实例的影响

应用远程监督技术来自动标注自然语言文本语料在解决了大规模语料的标注问题的同时，也给关系抽取模型带来了很多噪声实例，从而影响关系抽取的效果。为了缓解噪声实例带来的性能下降问题，本文采用了选择性注意力机制[42]。

1. 关系图谱的可视化

本课题重点是针对已有关系抽取模型存在的局限性，提出新的模型以提升关系抽取的效果。此外还将提出的模型应用到了科研文档中，为科研管理者提供一个语义检索系统。关系抽取模型的结果是三元组的形式，在用户检索后，这种关系事实以什么样的形式呈现就显得比较关键。本文采用的是D3js技术来对用户检索的关系图谱结果进行可视化。

本课题深入探索并分析了国内外研究学者在关系抽取方面的研究工作，发现目前已有的研究方法存在很多的局限性，这些正是本文要攻克的难点所在，主要分为以下两点：

1. 无直接关系触发词的句子的关系抽取

在自然语言文本语料中，普遍存在类似图1-1第一个句子的情况，即句子中不存在要抽取关系的直接触发词，而已有的关系抽取模型在这种情景下往往是无法抽取出有效关系的。

1. 长而复杂的句子的关系抽取

对于图1-1第二个句子这样长而复杂的句子，由于两个实体之间的距离隔得比较远，现有的方法对实体对上下文语义关系的学习比较困难，也常常不能提取出这类句子中实体之间的正确关系。



图1-1 关系抽取模型例子

## 研究内容及意义

### 研究内容

本课题研究的关系抽取是信息抽取的一项核心任务和重要环节。信息抽取旨在从大规模非结构化或半结构化的自然语言文本中抽取结构化信息，其主要任务是识别句子中的实体、抽取实体之间的语义关系以及其中涉及的事件[49]。关系抽取主要是从自然语言文本语料中抽取实体对的语义关系，然后表示成三元组的形式以构成关系事实，如：<Jobs, founder, Apple>表示乔布斯是苹果公司的创始人这样一个关系事实。这些关系事实是构建其他上层应用如语义检索、问答系统等的重要基础[50]。本课题研究内容的应用范围并不局限于科研文本，对于包括新闻文本、百科文本等在内的文本数据也同样是适用的。

对于关系抽取的基础性研究，为了方便与其他研究学者论文中的方法进行比较，本文采用的是NYT公开数据集。本课题的研究内容主要有以下几点：

1. 解决无直接关系触发词的关系抽取的困境

现有的关系抽取方法通常采用词嵌入[51][52]及位置嵌入[53]作为模型的分布式表示层，而这种表示存在1.2.2小节所述的第一个难点，即无法有效的抽取无直接关系触发词的句子中的关系。为了解决这一困境，本文提出实体类型嵌入(ETE)。ETE和词嵌入、位置嵌入一起作为模型的联合嵌入层。

1. 处理长而复杂的句子的关系抽取的问题

在应对长而复杂的句子时，现有的模型也常常不能提取出正确的关系，如图1的第2个例子。我们认为，以上抽取失败是由于缺失深层语境下的隐藏信息导致的。为解决这一问题，我们在提出的ETE的基础上，精心设计了循环分段残差网络(RPRN)结构。RPRN网络结构深而复杂，能够有效编码句子上下文中的潜在信息。

1. 构建语义检索应用系统

本课题重点是进行关系抽取的基础性研究，以解决当前已有方法所面临的困境。此外，也将提出的模型应用到了科研文档中，构建一个语义检索系统。检索的结果以关系图谱的形式呈现，这主要用到D3js可视化技术。

### 研究意义

本课题深入探索并分析了国内外研究学者在关系抽取方面的研究，提出ETE，并精心设计了RPRN网络结构。本文提出的模型有效解决了当前关系抽取研究长期以来面临的困境，如无法很好地抽取无直接关系触发词或长而复杂的句子中的实体对关系。本课题开展的这一基础性研究，极大地提高了关系抽取的整体性能，这对关系抽取的研究与发展具有重要的意义。

通过机器学习相关技术自动抽取自然语言文本语料中蕴含的关系事实，人们可以更加有效地获取文本中涉及的各主客体之间的语义关系，有效地避免了以人工阅读的方式从不断生成的海量文本中提取关系事实等结构信息，这对自然语言文本语料的信息化建设具有重大的意义。同时抽取的关系事实是构建语义检索、问答系统等上层应用的重要基础。

本课题提出的关系抽取模型，不仅对本文要构建的语义检索系统提供了基础，而且在一定程度上为今后关系抽取方面的研究指明了方向，本课题的研究具有广泛的实际应用和较高的理论研究价值。

## 本文的组织结构

本文共计包含六个章节，各章节的内容简述如下：

第一章：绪论。在本章节中，首先，简要介绍了本课题的背景及来源；其次，概述了国内外研究学者在关系抽取方面的研究，并剖析了关系抽取当前所面临的困境及本课题要解决的重难点问题；然后，简明扼要的阐述了本课题研究的主要内容以及研究的意义，最后描述了下本文的组织结构。

第二章：相关理论及技术概述。本章主要介绍了在进行关系抽取研究中涉及的关键理论，以及在建模过程中用到的核心技术。包括远程监督、循环神经网络、残差网络、选择性注意力机制等重要理论，还有深度学习框架tensorflow、知识图谱可视化框架D3js等相关技术。

第三章：关系抽取分析与设计。本章深入分析了关系抽取研究中遇到的重点及难点问题，并针对现有方法遇到的困境，提出ETE并设计RPRN网络结构。然后概述模型的总体架构，最后详述模型的整个设计过程。

第四章：关系抽取实现。本章节首先针对前一章节分析和设计的模型给出了具体的实现细节，包括关系抽取模型的算法实现过程及语义检索系统的关键实现。

第五章：模型评估及案例分析。本章首先简单介绍本文所设计的模型的超参数的设置及模型评估的相关指标，然后，以表格和曲线图的形式对我们提出的模型和现有的模型在各项评估指标上进行比较，最后，对模型进行关系抽取的不同结果做一个具体的案例分析，以佐证本文所提出的模型的有效性。

第六章：总结与展望。本章节简要总结了本课题开展的关系抽取的研究工作，并对其未来的发展和研究做了进一步的探讨和展望。

# 相关理论及技术概述

本课题开展的关系抽取这一基础性研究，主要基于机器学习相关理论与算法来建模。本文是在NYT大规模公开数据集上进行的实验，首先为了解决人工标注自然语言文本语料耗时耗力的问题，本文应用了远程监督。利用循环神经网络(RNN)[54]进行序列学习的优势，以及残差网络[55]深度编码模型表示的能力，本文设计了一种新的神经网络结构RPRN作为关系抽取编码器。为进一步增强模型在测试数据集上的泛化能力，在关系抽取建模过程中还引入了对抗训练[45]。

关系抽取的建模依赖于命名实体识别(NER)[56]任务，本文采用的是StanfordNLP[57]工具来完成这一任务。在关系抽取的设计过程中，我们应用了深度学习的相关方法，具体采用的框架是google推出的Tensorflow[58]。此外，在应用本文提出的模型构建语义检索系统时，对抽取的三元组关系事实采用Neo4j[59]图形数据库来存储，语义检索结果利用D3js[60]框架来进行关系图谱的数据可视化。

## 远程监督

关系抽取就是要抽取句子中包含的实体之间的语义关系，然后表示成如<Jobs, founder, Apple>这样的三元组形式。在传统有监督的关系抽取中，首先要对训练语料进行人工标注，因为关系抽取器模型本身是无法给要训练的实体对关系打标签的。然而，人工标注语料存在耗时耗力的问题，这极大地限制了大规模语料在关系抽取方面的应用。因此，Mintz等人[17]提出应用远程监督来自动标注语料。

### 远程知识库

应用远程监督来自动标注文本语料需要一个远程的知识库(KB)[61]。知识库是一种用于存储被计算机系统使用的复杂结构化或非结构化信息的技术[62]，包括专家系统所使用的规则集，文本中蕴含的关系事实、涉及的事件等等。本课题研究的关系抽取采用的远程监督知识库为FreeBase[20]，如下表2-1所示：

|  |  |  |
| --- | --- | --- |
| **relation** | **subject\_entity** | **object\_entity** |
| /people/person/nationality | Sanath Jayasuriya | Sri Lanka |
| /location/location/contains | Russia | Arkhangelsk |
| **…** | **…** | **…** |

表2-1 FreeBase知识库-关系事实

### 命名实体识别

命名实体识别(NER)是自然语言处理中一项非常重要的任务[63]，也是本课题要开展的关系抽取研究的重要基础。NER的主要目的是识别自然语言文本中的人名、地名、机构名、日期时间、专有名词等具有特定意义的实体。在关系抽取中要应用远程监督，首先就要识别出文本语料中的实体对，然后再结合知识库来自动标注。NER识别句子中的实体对如下所示：

**Steve Jobs** was the co-founder and CEO of **Apple** and formerly **Pixar**.

在上述实例中，使用NER可以识别出三个实体，这三个实体在该句子的上下文中可以组成两对实体，接下来就是自动标注实体对在对应实例中的语义关系。

### 实体链接

识别出了自然语言文本语料中涉及的实体后，就要采用远程监督来自动标注训练语料。在自动标注语料时，需要应用实体链接[64] ，如图2-1所示，即将训练语料中所提及的实体对链接到大规模远程KB中。

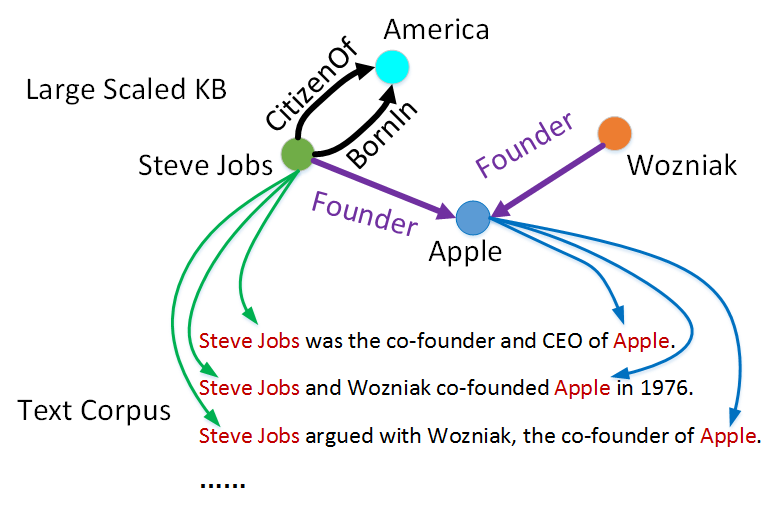


图2-1 远程监督实例

远程监督假设为：一个句子包含的实体对蕴含了该实体对在知识库中对应的关系，然后在模型训练时，将该句子作为其实体对对应关系的训练正例。最初的远程监督假设一个实体对只对应一种关系，但在自然语言文本语料中，大量存在一个实体对拥有多种关系的场景。因此，后续有很多的研究学者从不同角度对远程监督进行了改进，以适应其在大规模语料上的应用。

## 循环神经网络

循环神经网络(RNN)是一种处理序列数据的网络结构[65]，例如处理可变长度的自然语言文本序列。相较于全连接的前馈神经网络只有相邻层之间的神经元有连接而言，RNN的网络结构(如图2-2)的每一个隐藏层中的神经元也有连接，这使得RNN在几个时间步内可以共享相同的参数，有效避免了在不同位置上重复学习相同的特征规则。

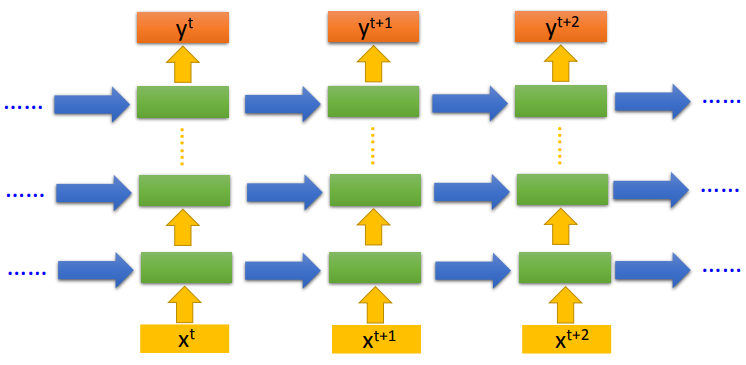


图2-2 多层循环神经网络

对于任何一个图灵可计算的函数而言，都可以通过图2-2这样一个有限维的RNN结构来模拟计算[65]，从这一角度来讲，RNN可以说是万能的。在RNN的设计过程中，主要包含以下几个重要的设计模式：

1. RNN模型每一个时间步都有输出，并且隐藏层神经元之间有如图3所示的循环连接；
2. RNN模型每个时间步都会产生一个输出，只有从一个时间步的输出到下一个时间步的隐藏层神经元有循环连接；
3. RNN模型隐藏层神经元之间存在循环连接，其读取整个序列后产生单个输出。

### 通过时间反向传播

反向传播(BP)算法自20世纪70年代被引入以来，作为一种神经网络模型的优化算法，一直是深度学习的主力[66]。而RNN是一种具有长期记忆能力的序列学习网络模型，普通的BP算法并不适用于这种模型的优化。在RNN模型中，采用的是通过时间反向传播(BPTT)[67]的神经网络优化算法，这是一种更有效的关于在RNN中传递参数的方法。在RNN的前向传播中，以特定的初始状态为起点，对于每一个从t=1到t=T的时间步，应用如下方程式来进行更新：

(2-1)

(2-2)

(2-3)

(2-4)

其中对于每一个时间步t，表示输入，表示隐藏层，表示输出，为模型输出向量，W、U、V表示权重矩阵，b、c为偏置向量。模型的损失函数为：

(2-5)

为在给定到t时刻的序列及t时刻的目标下的负对数似然，模型的总损失是所有时间步损失的累积之和。通过BPTT来计算RNN的梯度以下公式所示：

(2-6)

(2-7)

(2-8)

(2-9)

(2-10)

(2-11)

为对角矩阵，是关于t时刻与隐藏层神经元i关联的Jacobian，指的是包含这个元素。

### RNN变体

在普通RNN模型的训练过程中，存在一个众所周知的问题，及梯度弥散和梯度爆炸[68]。梯度弥散和梯度爆炸本质上可以归结为梯度不稳定问题，其主要原因是神经网络在优化时进行反向传播的过程中，同一矩阵经过了多次相乘，梯度倾向于消失或超过某个阈值，导致后续的神经元无法更新参数，神经网络的优化无法正常进行。

为了解决RNN在训练过程中遇到的上述问题，一些学者提出了RNN相关的一些变体结构。如非常经典的LSTM，其结构原理如图2-3所示。



图2-3 LSTM结构原理图

LSTM结构设计的巧妙之处在于：除了RNN拥有的外部循环外，内部结构还有自循环，这种结构具有学习长期依赖的能力。其主要由3个门来控制，分别为输入门，遗忘门，输出门。结构内部各状态的输出如下公式所示：

(2-12)

(2-13)

(2-14)

(2-15)

(2-16)

(2-17)

其中、、为输入门的权重矩阵，、、为遗忘门的权重矩阵，、、为输出门的权重矩阵，、、分别为3个门的偏置向量，为t时刻隐藏状态的输出向量。

LSTM的结构看起来比较复杂，那么该结构的所有构成都是必须的吗？有没有一种机制可以动态控制模型的遗忘功能呢？相较于LSTM，GRU[69]是一种更加简化的门控循环神经网络结构， 其结构原理如图2-4所示。

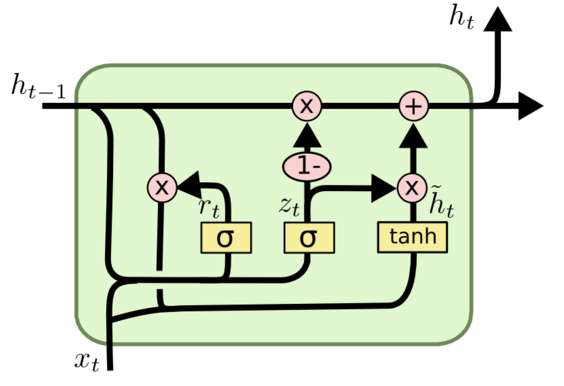


图2-4 GRU结构原理图

从图2-4与图2-3的对比中可以看出，GRU在原LSTM结构的基础上将门控单元简化成了2个，即更新门和重置门，公式表示为：

(2-18)

(2-19)

其中，、表示更新门的权重矩阵，、表示重置门的权重矩阵，和分别表示两个门的偏置向量，隐藏层状态向量为，由以下公式表示：

(2-20)

(2-21)

GRU结构已经不再显示包含记忆单元，但它至少和LSTM一样有效，并且收敛速度更快。与LSTM的最大不同在于，GRU中的单个门控单元可以同时控制遗忘功能和状态更新。

LSTM、GRU等门控机制的RNN模型都能够有效解决梯度弥散的问题，但处理不了梯度爆炸的问题。梯度截断(Gradient clipping)是一种非常流行的处理梯度爆炸问题的方法[70]，该方法通过截断梯度来使得传递的梯度总是小于给定的阈值。

## 残差网络

深度学习冠之以“深度”一词，以区别于早期浅层的神经网络模型。浅层网络模型最大的缺陷在于其很难学出具有不同抽象级别的数据表示，而深层模型则通过堆叠多层次网络结构有效地解决了这一难题。然而，越深的网络模型越难以训练和优化，甚至有时候训练出的效果还不如浅层模型。

为了解决深层模型训练及优化难的问题，Kaiming He等人[55]提出了残差网络(ResNet)模型架构。作者使用了深度达152层的ResNet在ImageNet数据集[71]进行评估，虽然该网络结构的深度是VGGNet-19模型的8倍，但其复杂度却比VGGNet-19低，并且在ImageNet测试集上的效果达到了3.57%的错误率，从而赢得了ILSVRC和COCO 2015比赛[[1]](#footnote-1)中多项任务的第一名。

ResNet的基本单元是卷积池化模块，然后以此为基础进行残差学习，进行残差学习的网络结构由恒等映射块来构造，块中的单元通过捷径来连接。

### 深度卷积网络

深度卷积网络在图像分类方面取得了一系列的突破[72]，其主要由一系列的卷积和池化两种操作构成。卷积的特性是输入表示的潜在模式比整个输入小且出现在不同区域，池化的则是通过再次抽样来获取主要特征而不失去原始输入的表征。

卷积操作的原理示意图如图2-5所示，每一层的卷积操作都会对应一组过滤器，然后用该组过滤器中的每一个分别与上层输出进行点乘操作。通过不同层次的卷积操作，输入表示被编码成了具有不同抽象级别的局部潜在特征。

卷积操作之后会进行池化操作，即将一层神经元簇的输出组合成下一层的某个神经元。池化操作的类型有多种，如最大池化、平均池化等。顾名思义，最大池化就是使用前一层每一个簇的最大值作为该层当前神经元的值，其操作原理如图2-6所示；以此类推，平均池化即使用前一层每一个簇的平均值作为该层当前神经元的值。当然，卷积和池化操作后还会经过激活函数处理，如sigmoid、ReLU[73]等激活函数。



图2-5 卷积操作原理



图2-6 最大池化操作原理

### 残差学习

假定将作为一个由几个堆叠的层(不一定是整个网络)匹配的潜在映射，其中是这些堆叠层的第一个的输入。假设多个非线性层可以渐进近似复杂的函数，那么等价于假设其可以渐进近似残差函数，即：

(2-22)

在以上公式中假定输入和输出具有相同的维度。并不是要堆叠层去近似，而是显示地让这些层近似，原始函数因此而变成了。尽管两种形式都要能够渐进近似期望的函数，但学习的难度却有很大的不同。残差学习不仅训练优化的复杂度低，而且效果极好。残差学习构建块原理示意图如图2-7所示。

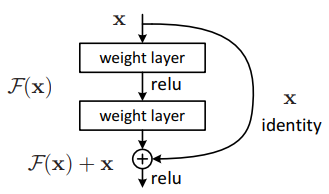


图2-7 残差学习构建块

### 通过捷径的恒等映射

上述提法可以通过拥有快捷连接的前馈神经网络来实现。快捷连接，即在构建块中跳过一或多个层。在残差学习中，快捷连接就是执行简单的恒等映射，然后将该层的输出添加到下一堆叠层的输出中。如图2-7所示的残差学习构建块可以定义为以下形式：

(2-23)

其中和分别表示所考虑层的输入和输出向量，表示要学习的残差映射，对于图2-7构建块中的2层，，为ReLU激活函数，为简化表示偏置项被忽略了。操作通过快捷连接和元素级相加来执行。公式2-23既没有引入额外的参数，也没有增加计算的复杂性，这不仅在实践中极具吸引力，而且在残差网络与普通网络的比较中也相当重要。公式2-23中的和必须具有相同的维度，如果不是这种情况的话，我们可以通过快捷连接执行一个线性投影来匹配维度：

(2-24)

当然，在公式2-23中也可以使用方阵，但对于处理退化问题来说恒等映射是足够的，故一般只用于维度匹配上。

### ResNet架构

ResNet基于普通的深度卷积网络，由堆叠的残差构造块组成，并在其中加入了快捷连接。ResNet架构与其他深度神经网络模型的对比如图2-8所示(左：VGG-19，中间：34层的普通深度卷积网络，右：34层的ResNet)。值得注意的是，和VGGNet相比，ResNet有更少的过滤器和更低的复杂度。

当输入和输出具有相同的维度的时候可以直接使用公式2-23，当维度在增加的时候，需要考虑以下两点：

1. 捷径依然执行恒等映射，再附加额外的零项填充以增加维度，这并不会引入额外的参数；
2. 公式2-24中的投影捷径用于匹配维度。

对于这两点而言，当快捷连接跨了两个大小的特征映射时，它们所执行的步幅都是2。

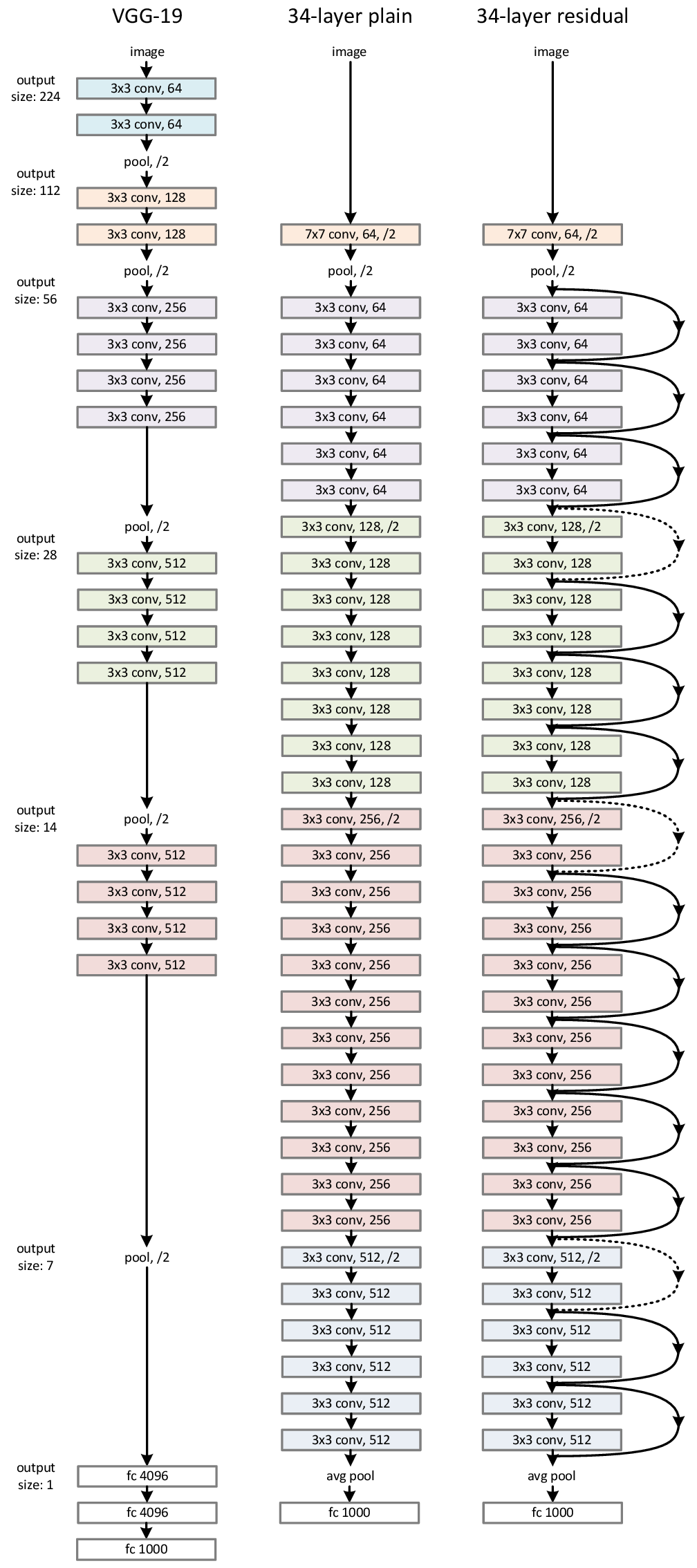


图2-8 残差网络与其他深度神经网络结构对比

## 对抗训练

尽管深度神经网络在各种不同的领域取得了巨大的成功，但其对输入数据中的噪声倾向于太过自信。对抗训练(Adversarial Training)就是一种通过生成对抗噪声到训练数据中的正则化分类算法[45]，以增强模型在测试集上的泛化能力。使用对抗训练编码实例的计算图如图2-9所示。

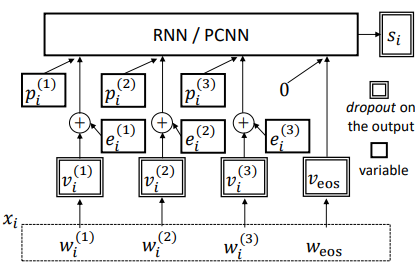


图2-9 使用对抗训练编码实例的计算图

其中表示第i个训练实例，表示关于的对抗扰动。对于一个实体对，关系的概率及损失函数表示如下：

(2-25)

(2-26)

其中为投影矩阵，为偏置向量，表示关系个数，为给定的参数集合。对抗训练会在训练过程产生持续的微小扰动，然后将产生的对抗噪声加入到词嵌入层。引入对抗训练后模型的损失函数表示如公式2-27所示，表示当前值的一个固定副本，因为从计算的角度来看，公式2-28是比较难处理的。

(2-27)

(2-28)

## StanfordNLP

StanfordNLP是斯坦福NLP组研发的一套NLP软件，它为主要的计算语言学问题提供了统计NLP、深度学习NLP和基于规则的NLP，这些工具将人类语言技术的需求结合到了应用中，并在工业界、学术界和政府中被广泛使用[57]。

本课题使用StanfordNLP来进行关系抽取模型训练前的数据预处理相关工作，主要包括用其中的POS Tagger工具对原始自然语言文本语料进行分词，然后用NER工具来识别语料中的命名实体，其示意如图2-10所示。

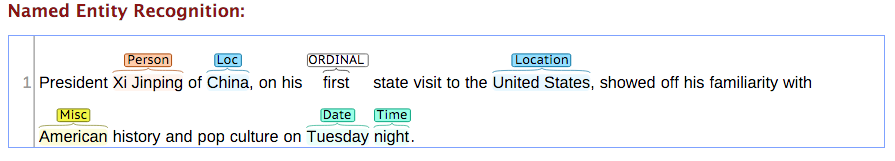


图2-10 命名实体识别实例

## Tensorflow

由google大脑团队研发的深度学习框架Tensorflow，是一种大规模、异构环境下运行的机器学习系统[74]，不仅大量用于机器学习方面的研究，而且在计算机视觉、语音识别、自然语言处理等计算领域被广泛应用。

Tensorflow使用数据流图来表示计算、共享状态以及改变该状态的操作。数据流图是由节点和边构成的有向图，节点代表计算操作，也可表示输入数据的起点或输出数据的终点，边表示两节点之间的输入输出关系。从Tensorflow的名字可以看出其最重要的两个概念：张量(Tensor)、流(Flow)。张量可以简单的理解为多维数组，流即为张量在计算图各节点上流动并通过节点上的计算而相互转化的过程。图2-11是Tensorflow中一个训练管道的数据流图的原理图，其中包含了用于读取输入数据、预处理、训练和检查点状态的子图。Tensorflow将数据流图的各节点映射到集群中多台机器上，其中一台机器跨多个计算设备，包括多核CPUs、通用GPUs和定制的被称为TPUs的ASICs，被分配到不同计算设备的各节点可以并行异步的进行计算。这种架构为应用程序开发人员提供了很大的灵活性：在其之前的“参数服务器”的设计中，系统内置了共享状态管理，使得开发者可以尝试新的优化和训练算法。

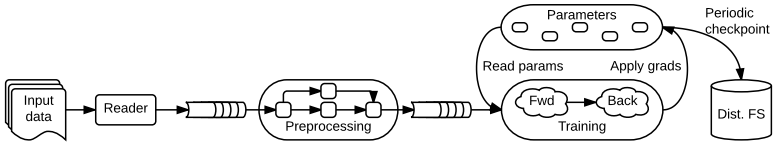


图2-11 Tensorflow中一个训练管道的数据流图

本课题开展的关系抽取研究利用深度神经网络来建模，至于具体的建模工具则采用目前最流行的Tensorflow。Tensorflow的计算图模型为本课题的研究及所提出的模型算法的具体实现提供了极大的便利和灵活性。

## 图形数据库

我们生活在一个彼此联系的世界，我们周围没有孤立的信息，只有丰富的、互相关联的领域。只有原生就包含关系的数据库才能有效地存储、处理、查询连接操作。在其他数据库通过代价很大的连接操作在查询时计算关系的情况下，图数据库将连接和模型中的数据一起存储。

简单的说，图形数据库就是一种将数据之间的关系和数据本身视为同等重要的数据库。其目的在不限制数据在预定义的模式下保存数据，相反，数据的存储方式与我们最初描绘它的方式是一样的，即显示单个实体是如何与其他实体进行连接或相互关联的。访问本地图形数据库中的节点和关系是一种高效、固定时间的操作，允许每秒钟快速地遍历每个核中的数百万个连接。

在进行关系抽取这一基础性研究的同时，本课题还将提出的模型应用到了科研文档的信息化建设中，构建了一个简单的语义检索系统。模型提取的关系事实是存储在图形数据库中，本文采用的是广泛使用的Neo4j图形数据库。

## 数据可视化

我们正处在一个信息洪流的大数据时代，那么数据以什么样的形式展现给用户就显得非常重要且意义重大。其中重点要考虑的是数据本身具有什么样的特点，展示的时候什么样的形式更加地简洁、清晰，以及怎样在用户获取到有价值信息的同时还会有一种美的享受。

考虑到本课题进行关系抽取的结果是三元组的关系事实，这些关系事实会构成一个庞大的实体关系网络，故本文在数据可视化技术选型时采用的是当前非常流行的D3js，然后再结合前端的HTML、CSS、JavaScript等技术，构建了一个以关系图谱的形式呈现的语义检索系统，呈现时可以与用户进行动态的交互。

## 本章小结

本章主要概述了在进行关系抽取的研究和实现过程中所涉及的关键理论和相关技术。首先，介绍了在关系抽取建模过程中基于的重要理论，包括用于自动标注大规模语料的远程监督，进行序列学习的循环神经网络，学习深层信息的残差网络，以及提升在测试集上泛化能力的对抗训练；其次，简述了用于关系建模的机器学习相关框架，包括用于自然语言处理的StanfordNLP，实现深度神经网络的Tensorflow框架；最后，简介了在进行关系抽取应用时所需要的相关技术，包括用于存储关系数据的图形数据库，进行数据可视化的D3js等前端技术。

# 关系抽取分析与设计

本系统在架构上采用了面向服务的思想，整个服务端分为服务提供端和服务消费端，使得系统充分解耦。服务提供端负责提供推荐服务，其中推荐引擎模型的设计是本系统服务提供端设计的重点，主要包括关系网络模型、推荐模型和推荐指标计算及存储方案等的设计。服务消费端和客户端主要是负责数据的清洗、数据组装和数据展现。

本章从系统的总体需求出发，分别分析了系统的服务提供端的模型需求和服务消费端的功能需求。着重介绍了系统总体架构设计、服务消费端的数据清洗模块、服务提供端的协作网络建模和协作推荐建模等的设计。

## 困境分析

本课题以合作者推荐为设计目标，通过分析和总结国内外推荐系统原理和应用案例中的不足，并结合推荐系统的设计流程梳理出本课题的需求。基于项目协作关系的合作者推荐系统旨在为解决科研人员寻找合作者效率和质量低下问题，同时为挖掘科研管理数据资源的潜在价值提供了新的方向。为广大科研人员寻找合作者提供了渠道，在一定程度上可以促进科研人员之间的交流，提升科研效率、科研成果水平。

## 模型架构设计

本推荐系统的总体架构设计充分考虑了系统的可拓展性、技术的前沿性、结构上的伸缩性和推荐服务模型的通用性。运用面向服务的思想将推荐服务模型抽象为通用的服务化组件，便于其他系统的接入，力求打造平台级的系统，为科研管理数据中心和全国科研人员更好的服务。本节将从功能架构和系统架构分别予以介绍。

## 联合嵌入

数据清洗模块主要包括数据来源及现状的分析、清洗规则的设计和清洗接口的设计。首先初步分析了数据来源及现状，说明了数据清洗的必要性；然后初步分析了关系网络图的特点，制定了一系列清洗规则和清洗流程，对清洗后的数据进行了初步的比对来说明清洗效果；最后设计了一套灵活的数据清洗接口。

### 词嵌入

本课题具有权威、丰富的全国科研管理数据资源， 本文选取了2010-2016年度全国科研管理数据中的项目立项数据和对应的人员数据，主要包括项目名称、项目立项年份、项目所属学科、项目参与者、作者姓名、作者性别、作者职称、作者单位等信息。其数据情况大体如下：项目立项数据共计24195条，与立项项目相关科研人员数据共计118187条。

为了初步分析科研人员之间的协作关系情况，本文以项目为中介，基于科研人员之间的项目协作关系，将科研人员和协作关系抽象为关系网络中的节点和边，建立无向无权网络，得到如下数据：

连通子图数9922；

最大连通子图节点数：73669；

图节点总数：118187；

连通子图节点数最多Top10 ：73669、106、86、74、70、70、68、67、65、62

通过以上数据可以看出：最大连通子图节点数占全图节点数的62.3%，而其余37.7%的节点却分布在9921个连通子图中。节点数较少的连通子图在全图中就像很多孤立的点，对于分析协作关系和合作者的推荐几乎没有贡献，而且还会造成计算和存储上的额外开销。

### 位置嵌入

由上面分析可知，协作关系网络中存在大量单点节和含节点数较少的小连通子图。因此，我们需要制定规则相应的清洗规则，用以去除在关系图中表现为单点和包含节点数较少的连通子图，从而减少计算和存储开销。

清洗规则主要分为以下三步：

第一步：取出给定年度范围内的立项数据；

第二步：根据取出的立项数据取出其相应的申报人和合作者，如果参与该项目的人数大于2人，则继续进行第三步；否则直接舍弃该项目及相应合作者；

第三步：判定该项目的申报人和合作者在给定年度范围内是否参与过其它的项目。如果至少有一人参与过其它项目，则保留该项目及人员数据；否则舍弃该项目及人员数据。

经过清洗规则清洗后的数据情况如下：

连通子图数量 1563

最大连通子图节点数： 73669

图节点数： 93045

连通子图节点数最最多top10：73669、106、86、74、70、70、68、67、65、62

通过与清洗前的数据对比可以发现节点数减少了大约20%，连通子图数减少了大约83%，而最大连通子图和包含节点数较多的连通子图基本没有变化。从而可以推断出减少的这些节点和连通子图都是包含较少节点的连通子图或是单节点。数据清洗后减少了计算和存储开销。

数据经过清洗后可以得到一些项目和人员信息，我们可以据此进一步从数据源抽取其他相关信息，具体流程如图3-4



图3-4 数据清洗流程图

### 实体类型嵌入

清洗后的数据组装成一定的格式并推送至服务提供端解析入库。这个过程中数据清洗是对基本数据的清洗过程，还需从数据源取出其他相关信息组装成数据仓库需要的格式。整个过程的数据组装处理都是在服务消费端完成的，数据格式组装完成后调用契约生成的统一接口将数据推送至服务提供端，由服务提供端解析数据并插入至数据仓库中。数据存储结构如下表所示：

表3-1 人员表user数据结构

|  |  |  |
| --- | --- | --- |
| 属性 | 数据类型 | 描述 |
| id | int | 主键，作者id |
| authorName | varchar | 作者姓名 |
| authorId | varchar | 生产环境数据人员id |
| gender | varchar | 性别 |
| title | varchar | 职称 |
| agency | varchar | 所属机构 |

表3-2 协作关系表 user\_relationship数据结构

|  |  |  |
| --- | --- | --- |
| 属性 | 数据类型 | 描述 |
| uId | int | 一作id |
| uName | varchar | 一作姓名 |
| uIds | varchar | 合作者ids,以;隔开 |
| uNames | varchar | 合作者姓名，以;隔开 |
| applicationId | varchar | 项目id |
| year | int | 协作年份 |

表3-3研究领域表user\_keyword

|  |  |  |
| --- | --- | --- |
| 属性 | 数据类型 | 描述 |
| uId | int | 作者id |
| uName | varchar | 作者姓名 |
| displine | varchar | 研究方向 |
| numbers | int | 该研究方向参与过的项目数 |

服务端之间的通信接口是契约统一生成的，其参数格式固定。服务消费端可以采用任何形式的数据源，包括数据库、excel表、xml、json等，只要组装成接口要求的数据格式推送到服务提供端即可。服务提供端也只需要接收固定格式的数据存储即可，而不管数据源的形式和来源，数据清洗接口示意图如图3-5所示。

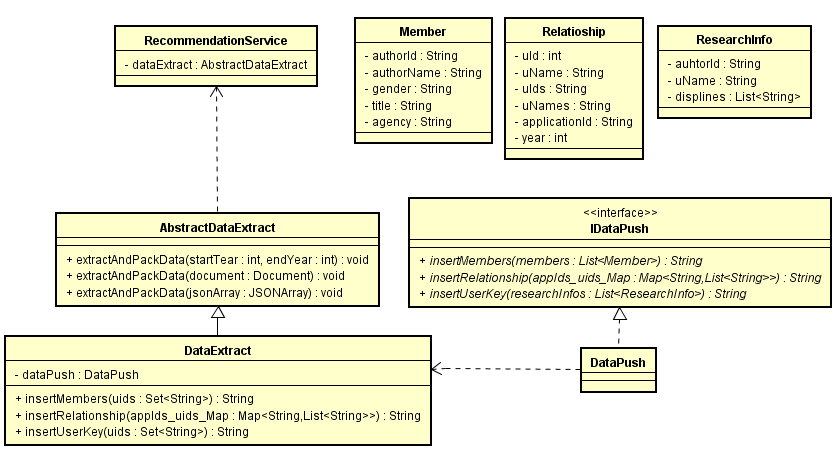


图3-5 数据清洗接口示意图

其中Member、Relationship、ResearchInfo等类定义了服务消费端和服务提供端的数据格式，IDataPush接口和DataPush类是服务消费端推送数据的接口和实现类，同时也是服务提供端处理数据的接口和实现类。以上这些接口和类都是通过Thrift契约定义的，通过Thrift工具可以生成对应的服务消费端与服务提供端接口及类。由于本系统服务消费端采用Java开发，服务提供端采用Python开发，故生成服务消费端接口和类均为Java类型，服务提供端需生成Python类型的接口和类。

AbstractDataExtract类给出了不同类型数据源的定义，由子类实现具体功能。extractAndPackData方法实现数据的清洗功能，子类继承AbstractDataExtract类并实现了数据组装功能，通过IDataPush接口推送数据至服务提供端。服务提供端实现IDataPush接口进行数据的存储操作。

RecommendationService拥有AbstractDataExtract子类实现的一个实例，为Action层提供服务。

## 基于RPRN的编码器

协作关系网络建模主要包含了网络模型图的选取、协作关系权重的设计和协作关系网络图的构建。首先分析了协作关系并将其抽象为两种网络图模型，分别分析了其特点；然后考虑了多维影响因子设计权重；最后利用合适的网络模型建立协作关系网络图。

### 序列学习

每一个社科项目都会有申请人和参与者，通常情况下有些申请人或参与者可能是其他项目的参与者或申请人，这些申请人和参与者的关系就形成了一个协作关系网络。网络中节点表示作者，边表示协作关系。文献[28]认为同一申请书中申请人与参与者之间存在边，其他参与者之间没有边的关联。文献[29][30]中对于合作关系的表示是只要有共同署名的项目就把他们视为合作关系，他们之间就有一条边。

现在假设立项项目《X1》有合作者，其中数字代表作者的顺序，为一作也就是项目的负责人或申请人。项目与作者之间的关系抽象出来如图3-6所示。按照文献[28]和文献[29][30]中对于合作关系的定义，可以把项目《X1》的合作关系抽象为图3-7和图3-8所示的网络图。



图3-6 项目X1与作者示意图

图3-7项目X1合作关系图 图3-8 项目X1合作关系图

图3-7星型耦合网络模型[14]表示的合作关系能够较好的区分第一作者跟非第一作者的关系，可以非常容易的体现出项目的核心成员，但却没有体现出非第一作者之间的关系。通常情况下，可以认为一个项目的申报者跟参与者之间的合作关系是直接的，但该模型却分割了非第一作者的合作关系。

图3-8全局耦合网络模型[14]表示的合作关系很好的体现出了第一作者与非第一作者的合作关系，参与过同一个项目的成员之间可以视为直接合作，这与现实情况十分吻合，但却体现不了不同作者之间的重要性和紧密程度地位。

由于本课题主要是分析项目作者之间的协作关系，通过以上两个模型的对比，可知图3-8全局耦合网络模型较为合适。图3-8中边的是由项目参与者按照协作顺序排列组合形成的，的次序分别代表协作顺序，那么网络图中的边有。虽然协作关系网络图为无向的，为了方便接下来边权重的定义，我们规定边中端点的协作顺序在端点的协作顺序之前。

为了解决全局耦合网络模型不能体现不同作者之间重要性和紧密程度的问题，本文把关系网络设计为无向加权网络，通过给定不同权重来区分作者之间的重要性和紧密程度，权重越大，重要性越高，联系越紧密。

### 残差网络

冯郁等人认为合作者之间不区分署名顺序，多次协作仅记录一次[29]。Hung-Hsuan Chen等人仅考虑合作者之间的协作次数，而不考虑合作者之间的署名顺序[30]。这些度量方式考虑的影响因素单一，并不能很好的衡量协作关系。

由多人共同完成的项目，一般而言第一作者和第二作者的贡献较大，联系也更加紧密；如果两个项目协作年份不同，那么我们有理由认为项目协作年份近的比项目协作年份远的作者之间的关系更紧密。基于以上事实，本文综合考虑了协作时作者的署名顺序、协作次数和协作年份等因素对协作关系的影响，设计了一套基于多维影响因子的协作关系权重度量方式。

一般而言第一作、第二作的贡献较大，联系也更加紧密；作者顺序越靠后，他们的重要性越弱，联系强度也更弱。我们有必要区别衡量项目协作时不同作者次序间的关联性强度。

定义两节点某次协作的权重为，其中分别表示某次合作时他们的作者排序，其表达式如下：

 （式3-1）

其中，为调节因子，且有，为常数，且有。

该表达式可以看出协作时作者次序越靠前，其权重越大，作者之间的协作关系更加紧密；协作时作者次序越靠后，其权重越小，作者之间的协作关系更加疏远。

一般而言，协作年份越近，说明当前协作关系越紧密；协作年份越远，说明当前协作关系越疏远。协作年份的远近跟协作关系衰减程度正相关，年份越远，协作关系的衰减越大。衰减因子的定义如下：

 （式3-2）

其中表示协作年份，表示当前年份，且有**，**为调节因子，一般可取20~50之间。根据该定义可知：指数不大于0，当且仅当协作年份等于当前年份时，其衰减因子为1，协作关系不衰减；当协作年份小于当前年份时，其衰减因子小于1，且距离当前年份越远，其衰减因子越小，协作关系衰减越快；为调节因子，越大衰减越慢，越小衰减越快。

在给定的年度范围内，有些作者会多次协作，定义作者对总的协作权重如下：

 （式3-3）

其中为作者协作的次数，为第次协作对应的年份。可以看出该权重定义综合考虑了协作时的作者顺序、协作年份、协作次数等因素，权重越大表明关系越紧密，再次协作的可能性也越大。该度量方式较为全面的衡量了多种因素对协作关系的影响，具有较高的参考性。

### 分段池化

协作关系网络采用的是无向有权图，构建协作关系网络的重点就是要计算出边信息和权重信息等。下面分别介绍了协作关系网络的融合过程、边信息和权重信息的计算。

按照3.4.1网络模型图的设计一节知，每一个项目的所有参与者都可以组成一个全局耦合网络。如果两个项目有共同的参与者，那么这两个较小的全局耦合网络就可以通过该共同参与者作为媒介连接成为一个较大的网络。如果这种项目有共同参与者的现象很多，那么很多小的全局耦合网络就会连接成为一个大的网络，最后组成的网络就会显得很紧密。

假设现有项目《X1》、《X2》，协作年份分别为，分别包含参与者和其中作者的次序代表他们协作时的作者顺序。按照3.4.1节网络模型图，两个项目的全局耦合网络和融合之后的网络图如图3-9所示。



图3-9 全局耦合网络和网络融合示意图

其中图3-9-a和图3-9-b分别为项目《X1》和《X2》的协作关系网络图表示。由于两个项目中有共同的参与者和，所以两个协作关系网络图可以通过参与者和作为媒介连接成为一个大的网络图，如图3-9-c所示。可以看出作为媒介的和在网络中拥有的边数较其他的节点多，从这一方面可以看出作者和在整个网络中协作的重要性。

同时对每个项目参与者在网络图中所组成的节点对计算协作权重作为该节点对组成边的权重，对于在多个网络中出现过的边，由于网络是无向的，把它们的累加权重作为最终边的权重。

对于节点在项目《X1》中分别对应第一作者和第二作者，在项目《X2》中分别对应于第二作者和第一作者。图3-9-c中边的权重是由图3-9-a中边和图3-9-b中边权重累加而成。

根据式3-1可知，对于图3-9-a有，对于图3-9-b有。假设项目年度范围中的项目数据结束年份为，根据式3-2有，项目《X1》协作关系衰减因子，项目《X2》协作关系衰减因子。由式3-3有融合后的网络图3-9-c中边的最终权重为：



同理，可计算出其它所有边的权重。

有了节点信息、边信息和权重信息，就可以据此建立一个无向有权图，该网络模型表征了协作关系，又称为协作关系网络。

## 选择性注意力机制

推荐模型设计主要包括协作关系相似度模型设计、研究领域相似度模型设计和推荐指标模型的设计。通过对传统节点相似度度量方式的分析，提出了一种基于局部不定长路径协作关系相似度模型，该模型极大的改善了传统相似度度量方式维度不足和算法复杂度过大的缺点。同时本文提出了基于研究领域重合度和研究领域侧重度的概念和模型来计算研究领域相似度，该模型能够很好的反映研究领域相似度。

首先分别通过协作关系相似度模型和研究领域相似度模型计算给定作者和待推荐作者的协作相似度和研究领域相似度，然后将两个相似度融合为最终的推荐指标，根据推荐指标的大小进行排序推荐。推荐系统模型示意图如图3-10。



图3-10 推荐系统模型示意图

推荐模型数据存储结构如下表：

表3-4 推荐数据表user\_recommendation数据结构

|  |  |  |
| --- | --- | --- |
| 属性 | 类型 | 描述 |
| uId | int | 作者id |
| rId | int | 待推荐作者id |
| score | decimal | 推荐指标 |
| rel\_score | decimal | 合作关系相似度 |
| rea\_score | decimal | 研究领域相似度 |
| overlap\_ratio | decimal | 研究领域重合度 |
| cos\_ratio | decimal | 研究领域侧重度 |

## 本章小结

本章对推荐系统主要模块和模型进行了分析和设计。首先，对推荐系统服务提供端和服务消费端进行了分析；紧接着对系统的的整体架构进行了设计，主要包括功能架构和系统架构；其次，着重围绕着服务提供端的相关模型和算法进行了介绍和设计，包括数据清洗模块的设计、协作关系网络模型设计、推荐系统模型设计和推荐指标计算及存储方案的设计。其中，数据清洗模块的设计包括对数据来源及现状的分析、数据清洗规则的设计和数据清洗接口的设计等；协作关系网络模型设计主要包括网络模型选择、协作关系权重设计和协作关系网络模型构建等；推荐系统模型设计主要包括协作关系相似度模型设计和研究领域相似度模型设计，协作关系相似度模型主要是提出了协作关系的度量方式和算法——基于局部不定长路径的协作关系相似度模型，研究领域相似度主要是提出了研究领域重合度和研究领域侧重度的度量方式——基于研究领域重合度和研究领域侧重度的研究领域相似度模型。

# 关系抽取算法实现

在上一章中，详细介绍了推荐系统的总体需求，并从整体架构到具体模块和模型的设计进行了深入的分析，夯实了推荐系统实现的理论基础。基于上一章的分析和设计，本章将对推荐系统的整体环境和框架搭建到具体模块和模型的实现做了详细介绍。

## 开发环境搭建

1）开发工具与框架

推荐系统主要分为服务提供端、服务消费端和客户端。下面分别介绍推荐系统的开发环境，如表4-1所示：

表4-1 推荐系统开发环境

|  |  |
| --- | --- |
| 工具/环境 | 软件名称/版本 |
| 服务提供端IDE | PyCharm 5.0.3 |
| Python | Python 3.4 |
| 服务消费端IDE | MyEclipse 10.0 |
| JDK | Sun JDK 1.8.0\_25 |
| 应用服务器 | apache-tomcat-7.0.72 |
| 开发服务器 | apache-tomcat-7.0.72 |
| 服务器系统 | Cent OS 5.7 |
| 数据库 | Oracle 11gR2 |
| 客户端软件 | 市面上常见的浏览器 |

推荐系统框架配置如表4.2所示，其中前五项为服务消费端所需组件，后三项为服务提供端所需组件。

表4.2 推荐系统框架配置

|  |  |
| --- | --- |
| 框架/组件 | 版本 |
| Highcharts | 3.0.6 |
| D3 | 3.3.9 |
| Struts2 | 2.3.15 |
| Spring | 3.0.5 |
| Hibernate | 4.1.9 |
| Networkx | 1.11 |
| Numpy | 1.8.1 |
| Scipy | 0.15.1 |

2）负载均衡配置

根据3.5.4节推荐结果数据计算方案分析和设计，我们把推荐结果计算服务部署到5个服务节点上，其IP为192.168.88.176-192.168.88.180。由一台服务器作为反向代理利用Ngnix作为负载均衡，其IP为192.168.88.160。

其核心配置如下：

|  |
| --- |
|  |

## 数据清洗模块实现

数据清洗是建模的第一步，数据清洗模块实现主要包括数据清洗规则和数据清洗接口的实现，图4-1为数据清洗模块流程示意图。



图4-1数据清洗模块流程图

第一步：用户进入数据清洗配置页面，配置数据清洗的相关信息，包括数据开始年份和数据结束年份等。

第二步：服务消费端接受到请求和参数后开始按照数据清洗规则进行数据清洗。在数据清洗完成后，进行人员数据的抽取及组装、研究领域数据抽取及组装和合作关系数据处理等操作，最后将三类数据推送至服务提供端。

第三步：服务提供端接受到推送数据后，对数据进行解析并入库。

其中服务消费端数据清洗主要是数据清洗规则的实现，人员数据的抽取及组装、研究领域数据抽取及组装和合作关系数据是数据清洗规则的后续实现，为调用数据推送接口做的准备。各类数据按照定义好的格式组装好后，通过调用统一接口推送数据至服务提供端，由服务提供端解析数据并入库，整个数据清洗流程结束。

### 数据清洗规则

根据3.3.2节中的数据清洗规则，其数据清洗规则实现框图如图4-2所示。



图4-2 数据清洗规则实现流程图

详细步骤如下：

第一步：进入数据清洗配置页面，选择数据清洗的年度区间，即配置起始年份和结束年份。

第二步：客户端将带有参数的请求发送至服务消费端，服务消费端根据配置的参数抽取该年度范围内的立项项目id集合applicationIds。

第三步：遍历立项项目id集合applicationIds，如果遍历完成则整个清洗过程结束，否则进行第四步。

第四步：取出立项项目id，并由项目id取出该项目的所有参与者members。如果参与者数量小于2，则返回到第三步；否则进行第五步。

第五步：遍历参与者集合members，如果已经遍历完成则返回至第三步，否则进行第六步。

第六步：取出成员member并查询其参与过项目的次数。如果参与过项目次数为1，则跳转至第五步；否则存储memberId、项目id与members的关系。

为了保证筛选后人员id的唯一性，其中memberId是以Set<String>的结构存储的；为保存项目id与人员id的对应关系和人员在项目参与过程中的合作次序，本文采用Map<String,List<String>>数据结构存储，其中key对应项目id,List<String>存储项目对应的人员id,并且其次序跟合作次序一致。

### 数据清洗接口

根据图3-5数据清洗接口示意图可知，数据清洗接口主要包含数据抽取、数据组装、数据交互等接口，其中服务消费端主要实现数据清洗规则、数据抽取及组装等接口，服务提供端主要实现数据交互接口。数据清洗规则接口包含了数据清洗及相应数据的抽取实现；数据抽取及组装接口主要是将数据封装为定义的格式；数据交互接口是由Thrift契约定义生成的，主要是在服务提供端实现数据的解析和入库操操作，服务消费端直接远程调用数据交互接口推送相关数据至服务提供端；人员数据和研究领域数据格式均由Thrift契约定义生成。上一小节已经介绍了数据清洗规则的实现，本小节主要介绍契约定义、数据抽取封装接口和数据交互接口的实现。

1. Thrift契约定义

首先通过Thrift定义服务消费端和服务提供端的通信契约，然后根据契约生成各自的接口和类。Thrift契约server.thrift文件定义如下：

|  |
| --- |
|  |

struct ResearchInfo定义的是研究领域ResearchInfo类，struct Mebmber定义的人员Mebmber类，service Server定义的是通信接口。通过运行命令

thrift –gen java server.thrift

thrift –gen py server.thrift

可以分别生成Java端和Python端接口和类。其接口分别如下：

Server.java生成接口源代码：

|  |
| --- |
|  |

Server.py生成接口源代码：

|  |
| --- |
|  |

1. 数据抽取组装接口实现

数据抽取及组装接口主要包含人员数据抽取及组装和研究领域数据抽取及组装。由于人员协作数据需要人员数据入库后的人员id，故人员协作关系数据直接传递初始关系数据至服务提供端，由服务提供端进行数据解析、转换、封装和 入库操作。人员数据和研究领域数据抽取及组装实现框图如图4-3所示。



图4-3 数据抽取及组装实现框图

服务消费端在数据清洗阶段主要是抽取、筛选和组装数据，数据组装成约定的格式后就可以推送至服务提供端进行数据入库。

其中数据的清洗和筛选已经在数据清洗规则实现一节介绍过，此处主要介绍数据的组装和数据推送的一些实现。

a）人员数据组装及推送

经过数据清洗规则后，可以得到保留的人员id集合uids。我们根据该人员id集合uids抽取出相应的人员信息，组装成Member对象并加入到List集合。人员数据组装成功后推送到服务提供端进行数据入库。核心代码如下：

|  |
| --- |
|  |

b）协作关系数据推送

协作关系数据是由项目id和项目参与者id组成的Map集合，直接将该协作关系推送至服务提供端，由服务提供端进行二次数据抽取、转换和组装。核心代码如下：

|  |
| --- |
|  |

c）研究领域数据组装及推送

首先根据保留的人员id集合uids抽取出对应的项目所属学科的集合作为该人员的研究领域集合，并组装成契约定义的ResearchInfos类的形式，最后将其推送至服务提供端进行解析入库。核心代码如下：

|  |
| --- |
|  |

1. 数据交互接口实现

数据交互接口主要定义了服务消费端跟服务提供端进行通信的标准和数据格式,服务消费端利用Thrift框架提供的底层通信机制跟服务提供端进行交互，数据交互接口主要在服务提供端实现，其框图如图4-4所示。

 图4-4数据交互接口框图

服务提供端接收服务消费端推送的数据并进行相应的处理。数据交互接口的实现主要是在服务提供端实现的，包括人员数据、研究领域数据和协作关系数据等的解析和入库等操作。服务消费端通过Thrift框架提供的通信机制与服务提供端进行RPC通信。

服务提供端数据交互接口实现核心代码如下：

|  |
| --- |
|  |

各个接口均调用了server层的具体实现，server层对各个接口的具体实现就是解析数据，然后插入进数据库,最后返回是否插入成功等信息。这样服务提供端就可以不用关注数据来源，只要数据格式符合给定的格式就可以对外提供服务。

## 协作网络建模实现

协作关系网络建模实现主要包括数据初始化、协作关系权重的计算和协作关系网络构建的实现，其流程图如图4-5所示。



图4-5 协作关系网络建模流程图

数据初始化主要是实现了加载和解析协作关系数据，将项目的协作关系组装为特定格式，以便于生成网络边信息；协作关系权重计算主要是实现了边权重的计算；协作关系网络构建主要是实现利用已有的边信息和权重信息进行协作网络的构建。

### 协作数据初始化

在整个推荐系统建模和对外提供服务过程中会多次用到协作关系数据，为了减小数据库的读写压力和便于管理，我们把数据库中的协作关系数据提取出来，按照一定格式以文本文件的形式存储，以后要用到协作关系数据时直接读取文件并解析即可。其文本格式如下：



每一行数据都代表着一个项目的所有参与者和项目年份，“-”将整行数据分隔成了三部分，第一部分表示该对应的作者是项目参与者中的第一作者；第二部分由“;”分隔，代表的是项目参与者中的非第一作者，它们的位置次序代表了协作时的作者顺序，例如是第二作者，是第三作者；第三部分为项目协作年份，用于计算该项目的协作关系衰减因子。

协作关系数据初始化包含数据加载和数据解析，首先从文本文件中加载所有数据，然后遍历每一行数据，每一条数据经解析后都会计算相应边的权重并加入关系网络，所有的数据初始化完毕后建模就完成了，其实现流程图如图4-6所示。



图4-6 协作关系数据初始化实现流程图

第一步：加载文本文件中的协作关系数据datas。

第二步：遍历协作关系数据datas，如果还有数据则转至第三步；否则整个建模过程结束。

第三步：取出一条数据并解析，根据数据格式分隔数据后，将项目所有的参与者按照协作次序加入list集合。将list数据和项目协作年份传至权重计算和网络建模模块。

数据加载核心代码如下：

|  |
| --- |
|  |

数据解析核心代码如下：

|  |
| --- |
|  |

### 协作关系权重

数据加载过程中，根据作者集合生成边的信息计算边权重，并结合项目协作年份计算衰减因子，最后计算衰减后的权重，并把该衰减后的权重作为初步构建网络模型中边的权重。计算协作关系权重具体实现流程图如4-7所示。



图4-7 协作关系权重计算实现流程图

第一步：根据协作年份结合式3-2计算协作关系衰减因子，并根据作者集合authors生成边的信息，其中为作者对应的协作顺序，且有。

第二步：根据式3-1计算边的权重

第三步：根据式3-3计算边的最终权重，并将权重作为边(author[i],author[j])的权重初步构建网络。

计算权重核心代码如下：

|  |
| --- |
|  |

上述两个for循环就是边的生成规则：按照排列组合的方式生成边，且满足。然后计算边的权重，最后计算衰减后的协作关系作为构建网络的边权。

### 协作关系网络

在数据初始化以及协作关系权重等信息计算完成后，利用networkx的Python库就可以很容易的构建一个无向有权图，该无向有权图表征协作关系网络。其协作关系网络构建的实现流程图如图4-8所示。



图4-8 协作关系网络构建实现流程图

第一步：利用之前已经生成的边信息和边权信息作为输入，判断现有协作关系网络模型中是否存在该输入的边。

第二步：如果网络模型中存在该边，那么取出该边的边权，并重置网络模型中该边的边权为该边在当前项目中的边权和网络模型中该边的边权之和。

第三步：如果网络模型中该边不存在，那么直接把边信息和对应的权重信息加入到网络模型中。

协作关系网络构建核心代码：

|  |
| --- |
|  |

## 协作推荐模型实现

协作推荐模型的实现主要包含协作关系相似度模型和研究领域相似度模型的实现、推荐系统的推荐指标以及推荐结果计算与存储方案的实现，下面分别予以介绍。

### 协作关系相似度模型

协作关系相似度模型是以协作关系网络为基础建立的。首先，进行网络初始化，加载并解析协作关系数据构建协作关系网络；其次，对给定节点的三度范围内的节点进行预处理，将局部关系网络图表示为邻接表的形式；再次，遍历三度范围内节点，针对每一个节点与给定节点组成的节点对找出其路径长度在内的所有路径；最后，计算节点对的每一条路径的相似度累加和，将其作为节点最终的协作关系相似度。具体流程框图如图4-9所示。

 图4-9协作关系相似度模型流程图

由于网络初始化主要是协作关系网络的建模，其实现在4.3节有详细的介绍，本节主要介绍节点的预处理、全路径寻址和相似度计算的具体实现。

1. 节点预处理实现

节点预处理主要是取出给定节点的三度范围内的所有节点，并把该局部关系网络表示成邻接表的形式。局部关系网络表示成邻接表主要是为了满足全路径寻址算法的输入要求。节点预处理实现框图如图4-10所示。



图4-10节点预处理实现流程图

首先给定一个作者author，在关系网络图中寻找其邻接点即为一度节点，将其加入nodes集合。针对每一个一度节点，再寻找其邻节点，即为author的二度节点，将其加入nodes集合。针对每一个二度节点，再寻找其邻节点，即为author的三度节点，将其加入nodes。这个过程结束以后就得到了author的所有三度范围内的节点。

遍历author三度范围内的节点，取出节点node的邻接点并用字典表示为dict[node]=neighbors的形式。

核心代码如下：

|  |
| --- |
|  |

1. 全路径寻址实现

全路径寻址就是要找出两节点之间符合条件的所有路径。经过上述节点的预处理流程后，对于给定节点*node*的三度范围内的关系已经表示成为了邻接表的形式。在该三度范围内给定一个节点对，要找出从节点到节点路径长度不大于的所有可达路径，可以采取深度优先搜索结合回溯法进行寻址。其主要思想如下：

第一步：把当前节点加入路径节点集合*path*末尾，并做以下校验：①路径*path*长度是否大于给定限制路径长度，若大于则回溯至的上一层节点；②节点是否跟目标节点*end*相等，若相等则返回该条路径*path*；③节点是否为叶子节点，若为叶子节点则回溯至的上一层节点。若校验后没有返回值，则转至第二步。

第二步：取出节点的邻接点，校验节点是否在路径*path*中已经存在，若节点在路径*path*中已经存在，则说明存在环路，此时不对做任何处理，直接处理的下一个邻接点并把其当做新的节点；若节点不在路径*path*中，则把节点置为当前节点重复第一步。

第三步：当回溯至节点时，表明从节点至节点的某一邻接点的所有路径寻找完毕，将该返回的路径加入*paths*，以同样的方式处理节点的其他邻接点。节点的所有邻接点处理完毕后返回最终的路径*paths*。

图4-11所示为包含6个节点的网络图，假设需要寻找节点之间路径长度在3以内的所有可达路径，即，图4-11中虚线和数字标识寻找方式和顺序。



图4-11 全路径寻址示意图

首先将当前节点=放入路径*path*的末尾，做三层校验通过后将的邻接点作为当前节点做同样的处理，此时路径*path*中顺序存在节点。将节点的邻接点作为当前节点做同样的处理，此时路径*path*中顺序存在节点。将节点的邻接点作为当前节点做同样处理，发现在校验路径长度时其路径*path*顺序存在节点，其长度为4，大于限制长度3，故返回至上一层节点。此时取的邻接点（因节点已经包含在*path*中了，故不需重复处理）做校验时发现该节点等于，说明已经成功找到一条路径*path*，其中顺序存在节点，返回该路径*path*。由于节点的邻接点已经遍历完毕，故将返回的路径*path*加入到路径集合*paths*中，并返回至的上一层节点。接下来遍历的其它邻接点，此处即为节点，做校验时发现该节点跟节点相同，说明成功找到一条路径*path*,其中顺序包含节点，返回该路径*path*。由于节点的所有路径已经遍历完毕，故将返回的*path*加入路径集合*paths*，并返回至上一层节点。

对的其他邻接点，此处为做同样的处理，会找到一条路径*path*，其顺序包含节点。由于节点的路径已经遍历完毕，故将返回的路径*path*加入路径集合*paths*,并返回至上一层节点。

经过以上处理，节点的所有路径均已遍历完成，此时返回最终的路径集合*paths*。

其核心实现代码如下：

|  |
| --- |
|  |

第2行是把当前节点顺序加入*path*变量

第3-8行是做的三层校验，分别校验了长度、是否成功找到路径、是否是叶子节点。满足校验条件就会返回至上一层节点进行下一次递归。

第9行是当前递归的路径集合变量，用来存储经过当前节点的所有符合条件的路径。

第10行是处理当前节点的邻节点。

第11行保证了不会形成环路

第12行把当前节点的邻接点视作当前节点进行递归操作

第13-15行把返回的符合条件的路径加入路径集合变量*paths*.

第16行当前节点的所有邻接点遍历完毕就返回经过当前节点的路径集合*paths*作为上一层的路径集合的子集。

1. 相似度计算实现

计算给定*author*和节点*node*的协作关系相似度就是要使用以上全路径寻址算法找出两节点间路程长度在内的所有路径*paths*，然后根据3.3.1节提出的相似度模型计算所有路径的相似度累加和作为两节点协作关系相似度。具体实现如图4-12所示。



图 4-12协作关系相似度实现流程图

第一步：初始化相似度值。

第二步：遍历两节点之间的路径*paths*，如果路径遍历完毕则直接返回相似度；否则进行第三步。

第三步：取出路径*path*，令下标，单路径相似度r=1。

第四步：判断下标是否越界，若越界则单路径r相似度累加至最终相似度s上，并转至第二步；否则取出相邻两节点。

第五步：根据式3-3计算边权，根据式3-4计算权重，根据式3-7计算单路径相似r，同时下标i加1。转至第四步。

### 研究领域相似度模型

由3.5.2节知，节点对的研究领域相似度由其研究邻域重合度和研究领域侧重度融合而成，研究领域相似度实现流程图如图4-13所示。



图4-13研究领域相似度实现流程图

对于给定的节点对，首先分别找出节点参与的所有项目所属学科组成的集合作为其各自的研究领域和，然后分别计算出其研究领域交集和并集。

按式3-9计算研究邻域重合度

按式3-10计算研究领域重合度

分别求出的统计均值。将统计均值代入式3-17：可以求出，取，即有研究领域相似度为：

 （式4-1）

### 推荐系统推荐指标

推荐指标是由研究领域相似度和协作关系相似度融合而成，其实现流程图如图4-14所示。



图4-14推荐指标融合实现流程图

在3.5.3节中推荐指标综合相似度为：



当研究领域相似度时，有；

当研究领域相似度时，有

其实现核心代码如下：

|  |
| --- |
|  |

### 推荐指标计算与存储方案

在3.6推荐指标计算和存储方案设计一节分析了推荐系统的数据量，并从数据计算和数据存储方面给出了设计方案。本节主要分别介绍数据计算方案和数据存储方案的具体实现，其中数据计算方案的实现主要介绍多线程下推荐结果数据的计算；数据存储方案主要介绍了数据存储结构、数据入库和数据查询等的具体实现。

1. 数据计算方案实现

数据计算所需的平台搭建在4.1.2小节已经介绍过，该小节主要介绍多线程下推荐结果数据的计算实现。

推荐结果数据计算的核心代码如下：

|  |
| --- |
|  |

第一步：获取锁，以便保持后续操作的原子性。

第二步：传入节点id，根据推荐模型计算推荐结果信息，主要是找节点三度范围内的所有节点nodes，根据模型计算 id跟nodes组成节点对的协作关系相似度、研究领域重合度和研究领域侧重度。

第三步：组装数据。将上述通过推荐模型计算的数据组装成特定的格式以便后续批量插入数据库。

第四步：释放锁，以便其他的线程执行计算任务。

第五步：将以上推荐结果组装后的数据插入对应的数据库。

上述代码为推荐结果数据计算的底层实现，下面介绍多线程处理推荐结果计算，核心代码如下：

|  |
| --- |
|  |

第一步：接收推荐结果计算任务后，加载协作关系数据并创建协作关系网络图。

第二步：根据参数起始任务段来批量获取相应段的作者id。

第三步：创建线程并将推荐结果数据计算方法跟该线程关联，同时传入该段的作者id集合。

第四步：启动线程。

1. 数据存储方案实现

根据3.5.4节分析可知，推荐结果总数据量在690万左右(实际推荐结果总数约为680万)，为了提高存储效率和查询效率，我们采用了分表和数据库索引技术。

存储推荐结果数据的表结构和分表存储如表3-4和表3-6所示。其数据库中分表存储设计示意图如图4-15所示。

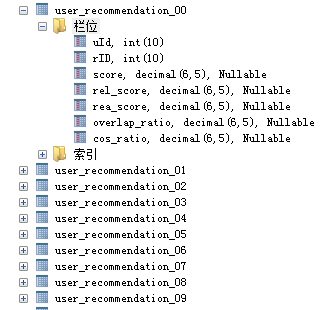


图4-15 推荐结果数据分表存储示意图

在计算不同任务段的作者推荐数据时，根据作者id所属段来将其存储至对应的分表中，其对应关系见表3-6，其对应关系核心代码实现如下：

|  |
| --- |
|  |

给定作者的推荐结果数据计算完成后就组装成为特定格式，然后进行批量插入，其核心代码如下：

|  |
| --- |
|  |

上述是作者id所属段，有。根据获取其对应存入的表名，然后执行批量插入操作。

根据3.5.4节分析可知，每一个分表的数据量在69万左右。如果在建表时不做任何处理，对推荐结果进行查询时每一次都需要全表扫描，这样查询操作非常耗时、效率极其低下。为了解决查询效率耗时和效率低下的问题，除了采用分表技术以外，本文还采用了数据库的索引技术。在每一个分表的uid字段上都建立索引，这样在通过uid字段查询推荐结果时就不需要进行全表扫描。需要注意的是，对分表uid字段建立索引是要在该分表所有推荐结果数据入库以后建立，如果在建立分表的时候就建立索引，那么每一次推荐结果的入库都会造成该索引的更新，使得索引的维护成本极高，从而降低了数据库的性能进而影响整个推荐结果计算的性能和存储的性能。

在分表uid字段建立索引代码如下：

|  |
| --- |
|  |

user\_recommendation\_00表的uid字段建立了索引，同理可以在其他分表的uid字段建立索引。

## 本章小结

本章详细介绍了基于项目协作关系的推荐系统的各个模块和模型的实现。首先，简单的介绍了开发环境与框架的配置；其次对数据清洗模块的实现做了详细的介绍，包括数据的清洗规则和清洗接口的实现等；然后介绍了协作关系建模的实现，主要包括协作数据初始化、协作关系权重和协作关系网络的实现；最后介绍了协作推荐模型的实现，主要包括协作关系相似度模型、研究领域相似度模型、推荐系统推荐指标模型和推荐指标计算与存储方案的实现，其中协作关系相似度主要分为网络初始化、节点预处理、全路径寻址和协作相似度计算等部分；研究领域相似度主要包括研究领域重合度和研究领域侧重度等；推荐指标主要是协作关系相似度和研究领相似度的融合等。

# 模型评估及案例分析

第三章和第四章详细介绍了推荐系统的设计和实现，本章将从不同的角度对推荐系统进行测试和分析，主要包括系统的性能测试及分析和协作关系网络数据分析两大部分。系统测试及性能分析主要是从系统的基本功能、响应时间和推荐质量来进行分析，协作网络数据分析主要是探索和分析协作关系网络呈现的一系列规律。为了便于描述，本章将采用图片和表格来进行展现和分析测试结果及相关数据。

## 系统测试及性能分析

测试主要是为了保证系统功能足够完整、系统足够健壮、用户界面友好并且响应时间合理。推荐系统的测试主要将采用黑盒测试的方式，同时辅之以白盒进行测试。测试内容主要包括以下几个部分：基本功能测试，即系统对外提供基本推荐功能的测试；响应时间测试，主要测试各推荐功能模块的响应时间；推荐质量分析，主要是对推荐质量进行测试。对于本课题的测试环境，客户端采用Chrome浏览器和Firebug调试工具，服务端采用Mysql数据库和Tomcat应用服务器。

### 系统测试方案

推荐系统的基本功能测试包括作者检索、合作者推荐、协作关系图展现、协作关系数据概况和系统内部功能数据抽取和推荐结果计算等，响应时间测试主要包括基本功能的响应时间，推荐质量分析主要是分析推荐的准确率、召回率和覆盖率。

用户进入合作者推荐页面，通过提供的搜索框检索待推荐的作者，返回检索结果，用户可以选择某一作者查看其推荐结果，推荐结果以列表的形式展现，待推荐者和推荐结果之间的协作关系以图的形式展现出来。用户进入协作网络数据概况导航栏可以查看协作关系网络数据概况。数据抽取和推荐结果计算主要是系统内部功能，配置好数据抽取年份后进行数据清洗、数据抽取、数据推送、数据解析和数据入库，推荐结果计算主要是采用了多机多线程的和分表的技术。

对系统基本功能和系统内部功能进行测试并记录下请求响应的时间。

### 系统测试结果分析

1. 基本功能与响应时间测试结果分析

根据5.1.1节中描述的测试方案对推荐系统进行测试，基本功能测试结果如表5.1所示。

表5.1基本功能测试

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 功能 | 输入与动作 | 预期响应 | 实际响应 | 是否通过 |
| 01 | 进入推荐页面 | 点击菜单栏 | 显示作者检索框 | 显示作者检索框 | 是 |
| 02 | 作者检索 | 输入作者并检索 | 显示检索的相关作者列表 | 相关作者列表展示 | 是 |
| 03 | 作者推荐 | 点击按钮触发推荐操作 | 展现推荐作者列表和关系图 | 推荐结果列表展示 | 是 |
| 展示作者关系图 | 是 |
| 04 | 数据概况展现 | 点击页面导航栏 | 展示相关数据图表 | 展示相关数据图表 | 是 |
| 05 | 抽取数据 | 配置参数提交 | 提示数据抽取成功 | 提示数据抽取成功 | 是 |

表5.1所示的测试结果显示，推荐系统的基本功能均正常运行。用户点击和输入操作也的到正常响应。表5.2为以上基本功能的响应时间。

表5.2 基本功能响应时间

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 功能 | 处理数据量 | 平均响应时间（ms） |
| 01 | 作者检索 | 数百条 | 32 |
| 02 | 推荐作者列表 | 数百条 | 92 |
| 03 | 推荐作者关系图 | 数百条 | 98 |
| 04 | 数据概况展现 | 数百条 | 24 |
| 05 | 抽取数据 | 数十万条 | 256000 |

除了数据抽取的耗时较长之外，其他响应时间均在100ms以内，极大的增强了用户体验，这是由于系统事先已经计算出了推荐结果，推荐服务会直接查询相关结果。由于数据抽取处理数据量较大，且涉及流程较多故其耗时较长，但是数据抽取只在数据更新后由系统维护人员进行操作，并不影响用户体验。

整个推荐系统耗时最大的就是推荐结果的计算，其处理数据量达到数百万条甚至数千万条，共计有近十万节数据节点，以3000数据节点为单位进行推荐结果计算测试，取多次测试的平均值作为在该不同环境下的测试的结果，测试结果如表5.3所示。

表5.3推荐结果计算耗时测试

|  |  |  |  |
| --- | --- | --- | --- |
| 环境 | 节点数 | 实际推荐结果平均值条数 | 平均耗时（h） |
| 单机单线程 | 3000 | 237500 | 18.25 |
| 单机多线程（2线程） | 3000 | 226197 | 12.62 |
| 集群多线程（2线程） | 3000 | 221965 | 2.53 |

根据表5.3计算结果可以看出通过使用集群多线程可以大大的提高计算效率，集群数和线程数取决于硬件资源。

图5-1至图5-3为推荐系统推荐结果可视化（部分敏感信息做打码处理）。



图5-1 待推荐作者检索结果

在图5-1中检索结果列表为用户键入的待推荐的作者相关信息，包含了作者姓名和作者所属机构用以区分同名的不同作者。最后一列是操作列，点击可以查看推荐给该作者的潜在合作者信息，如图5-2所示。

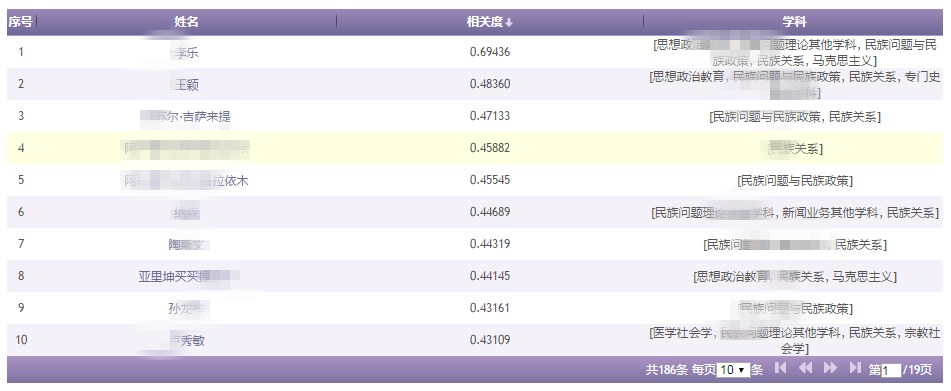


图5-2推荐结果列表

图5-2中为某作者的潜在合作者推荐结果，包含了姓名、推荐指标和相应作者的学科，其中按照推荐指标降序排列。

图5-3为某作者跟其潜在合作者在协作关系网络中的协作关系示意图，该图仅展示了该作者在协作关系网络中三度范围内的协作关系。最大的顶点为给定的作者，其他节点为推荐的潜在合作者。

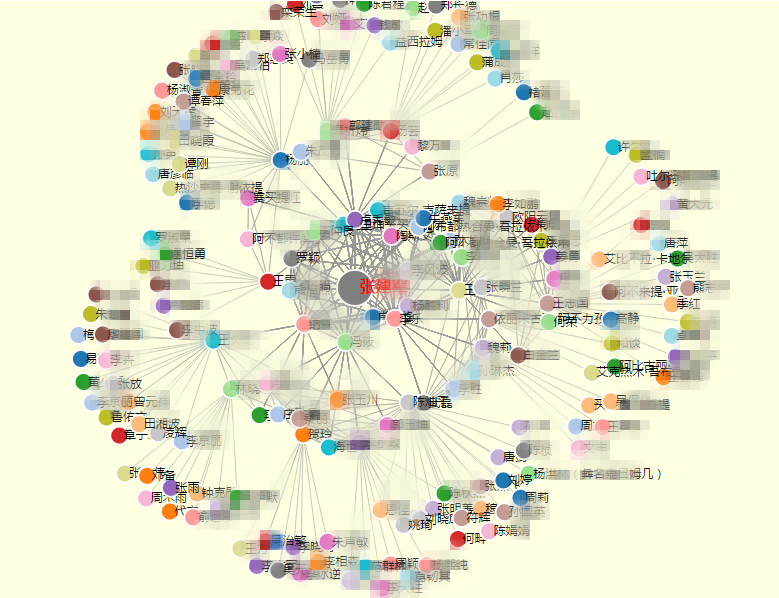


图5-3推荐作者协作关系图

1. 推荐质量分析

推荐质量主要从推荐准确率、召回率和覆盖率三方面分析[[[2]](#footnote-2)]，记推荐结果总人数为,其中推荐且成功合作的人数为，测试集总人数（合作总人数）为，推荐结果去重后总人数为，训练集总人数为，定义准确率，定义召回率为，定义覆盖率。

分别实现了基于协作关系相似度跟研究领域相似度的混合推荐模型（CNR）和基于共同邻域（Jaccard）跟研究领域相似度模型的混合推荐模型（CMR）作为对比，以2017年度社会科学数据为测试集，每一个节点返回推荐结果数以5递增，其模型准确率、召回率和覆盖率如图5-4、5-5和5-6所示：

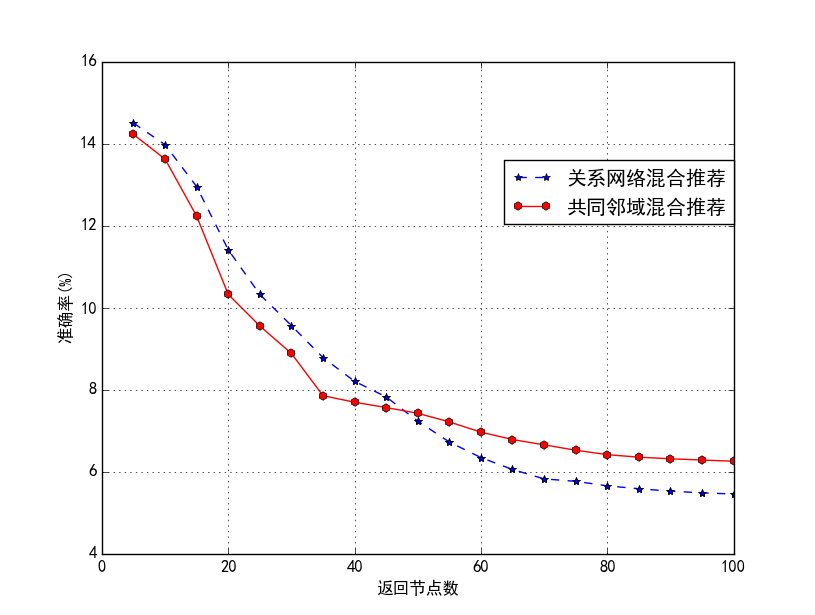


图5-4 准确率随推荐返回节点数的变化

由图5-4可知，在返回推荐节点数小于50时，本文提出的推荐模型准确率优于基于共同邻域的混合推荐模型，说明本文提出的模型在二度范围内具有较好的推荐性能；但在推荐节点数大于50时，基于共同邻域的混合推荐模型准确率优于本文提出的基于协作关系相似度跟研究领域相似度的混合推荐模型，且基于共同邻域的混合推荐模型准确率变化较小。这是因为基于共同邻域的混合推荐模型分析的是二度范围内的节点，当推荐节点数大于50时，这些节点大多数属于三度节点，而该模型处理不了三度节点，对最终结果影响较小，故呈现出准确率变化较小的趋势。另一方面，本文提出的模型具有较大的推荐范围，在返回推荐节点数大于50时，其对推荐准确率的影响较大（返回推荐节点数越多，其准确率会降低）。

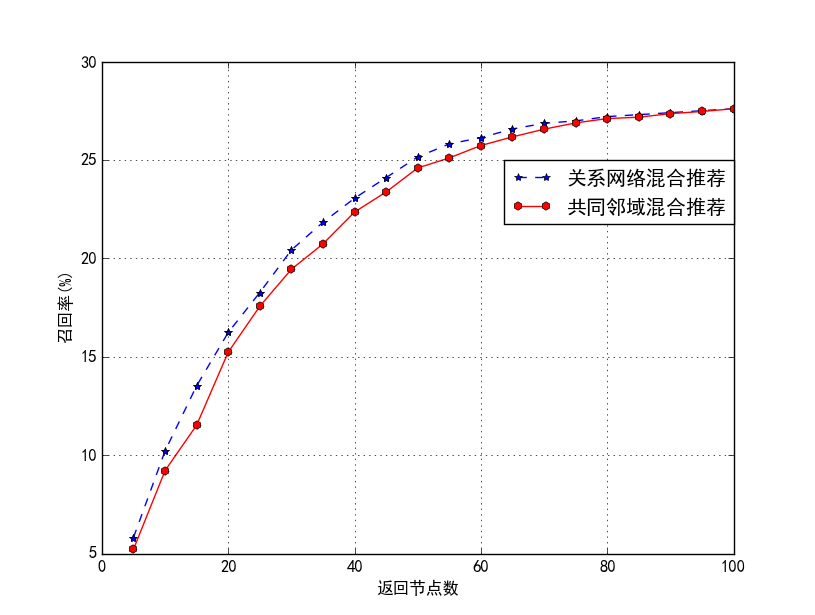


图5-5 召回率随推荐返回节点数的变化

由图5-5可知，召回率随着推荐总节点数的增加而增加，跟准确率的变化趋势相反，可以看出在返回推荐结果数相同的情况下，本文提出的模型召回率优于基于共同邻域的混合推荐模型。

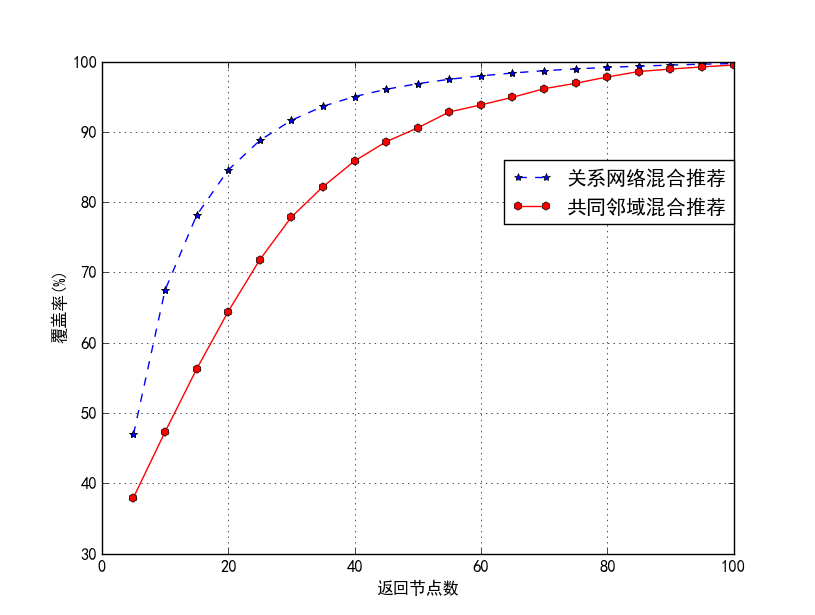


图5-6 覆盖率随推荐返回节点数的变化

由图5-6可知，覆盖率随返回推荐总节点数的增加而增加，初始增加幅度较大，最终趋于100%，本文提出的模型在返回节点数相同的情况下覆盖率优于基于共同邻域的混合推荐模型。结果表明返回推荐节点数越多，其覆盖率会越大，召回率越大，准确度越低。

综合以上三个指标可以看出，本文提出的基于协作关系相似度跟研究领域相似度的混合推荐模型总体要优于基于共同邻域（Jaccard）跟研究领域相似度模型的混合推荐模型。

## 协作关系网络探究

在整个推荐系统的设计和实现过程中，我们发现了协作关系网络呈现出的一些有趣的现象，包括协作关系概况数据、连通子图数据和度分布数据等呈现的一些规律，下面分别展示这些现象并尝试解释这些现象。

### 协作关系概况数据分析

协作关系概况数据主要包括协作关系网络历年全图节点数、最大连通子图节点数以及最大连通子图节点数占全图节点数的比例。其各年度数据示意图如图5-7和表5.4所示。

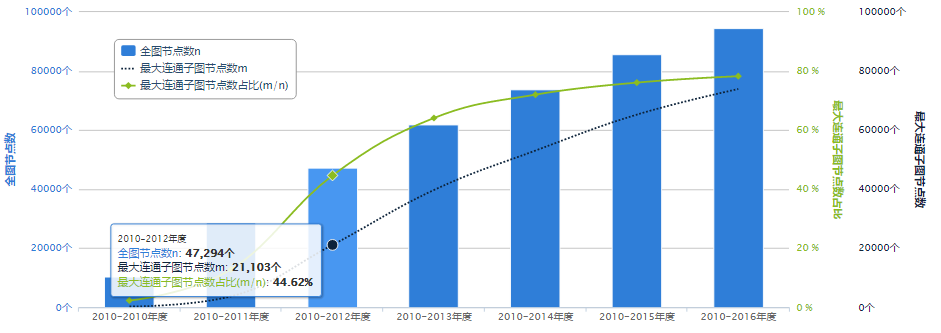


图5-7 协作关系概况数据图

表5-4协作关系概况数据

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 年度 | 全图节点数(n) | 最大连通子图节点数(m)个 | 最大连通子图节点数占比（m/n）% | 占比增量 |
| 2010-2010 | 10297 | 235 | 2.28 | 2.28 |
| 2010-2011 | 28724 | 3682 | 12.82 | 10.54 |
| 2010-2012 | 47294 | 21103 | 44.62 | 31.8 |
| 2010-2013 | 61946 | 39705 | 64.09 | 20 |
| 2010-2014 | 73688 | 53050 | 72.01 | 7.92 |
| 2010-2015 | 85737 | 65235 | 76.08 | 4.07 |
| 2010-2016 | 93045 | 73906 | 79.43 | 3.35 |

通过图5-7可以看出，随着全图节点数的增加，其最大连通子图节点数也在增加，但最大连通子图节点数占比的增量是先大后小。最大连通子图节点数增加是因为新加入的节点有部分是原来最大连通子图中已经存在的节点，这部分节点作为媒介将较小的连通子图融合到大的连通子图中，这样最大连通子图就会不断的增长。

全图节点数的变化和最大连通子图节点数都是呈现出先是增量较大后是增量较小的趋势，导致这一变化趋势的原因可能是：随着人文社会科学的发展，后面立项的作者很有可能在之前已经立项过，在统计作者时不会新增作者。所以随着数据的积累，如果以后立项项目中有很多作者之前已经立项过的，那么其作者的增量不会很大，且其作者增量越小，表明其立项项目越是集中在之前立项过的作者之中，进一步可以推测新一代的社会科学类研究人员科研成果增长缓慢。

### 连通子图数据分析

通过以上协作关系概况可以知道，每一个年度范围内的协作关系网络都存在一个最大连通子图，且该最大连通子图包含了全图绝大多数的节点。为了进一步探究连通子图的的变化规律，我们统计了每个年度范围内协作关系网络包含节点数前十的连通子图，如图5-8所示，横坐标表示子图节点数的排序，纵坐标表示该子图所包含的节点数。

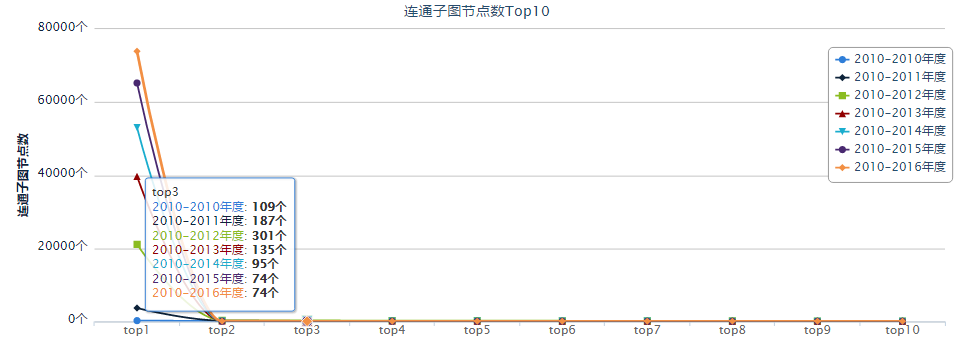


图5-8连通子图节点数Top10

通过图5-8可以看出，每一个协作关系网络中存在一个最大连通子图，并且也仅仅只存在一个巨片（包含节点数较多的连通子图），其他连通子图节点数都较少。另外，除最大连通子图之外的连通子图呈现出越接近当前年份其节点数越少的趋势。

随着社会科学项目数据的积累，新加入的项目参与者很有可能是已经存在于之前的数据之中的，这样在挖掘项目协作关系时，新加入的小的协作关系网络就会通过那些已经存在于协作关系网络中的节点作为媒介而融合成为一个较大的协作关系网络，而那些没有媒介的新加入的较小的协作关系网络就会表现为较小的连通子图，这就会造成具有巨片特征的网络节点数会越来越大。小连通子图在不断的生成，也在不断的被巨片“吞噬“而形成更大的巨片，这就会使整个网络呈现出少数巨片和众多小连通子图的现象。

巨片越少，且巨片节点越多，表明协作关系网络整体较紧密；反之表明协作关系网络整体较松散。图5-8所呈现的巨片仅有一个，且节点数较多，说明该协作关系网络较紧密，这也是我们分析协作关系的基础。

通过图5-8，观察到小连通子图呈现出越接近当前年份其节点数越少的趋势，并且子图节点数从较远年份到当前年份呈现出先增加后减少的趋势。为更好的证实和探究这一趋势，我们选取了每一个年度范围内协作关系网络包含节点数在前三十的小连通子图，如图5-9所示，横坐标为连通子图按节点数排序，纵坐标为该连通子图节点数。

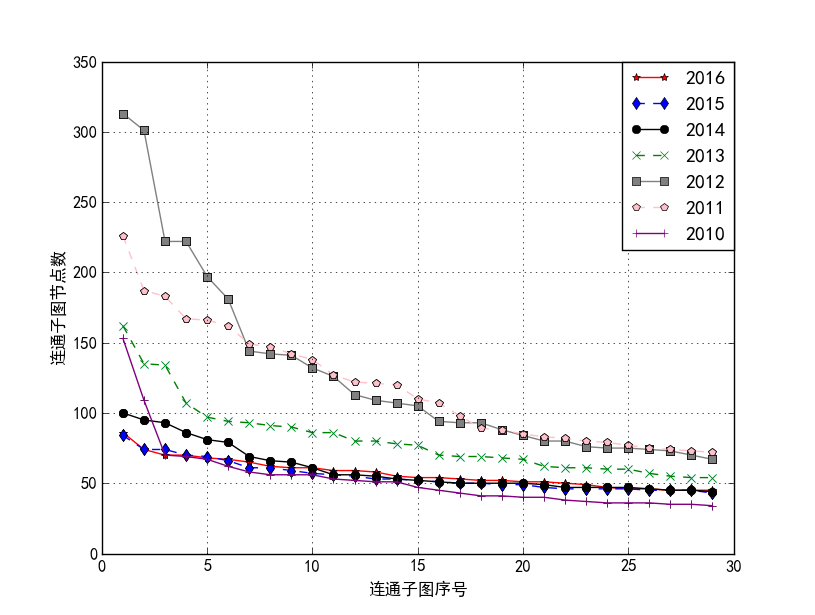


图5-9 连通子图节点数top2-top30

可以看出，2010-2012年连通子图节点数整体基本呈现递增的趋势，但到2013年及以后就呈现出递减的趋势。造成这一现象的原因可能是在数据积累初期，新加入的节点组成的较小连通子图由于缺乏媒介而无法融合到巨片中，就只能以较小的连通子图存在于网络中；随着数据的积累，新加入的节点存在于原来网络中的概率变大，也就是小连通子图跟巨片之间存在媒介的概率变大，也就更容易与巨片融合形成更大巨片，这就造成了一种连通子图被巨片“吞噬”的现象，且连通子图节点数越多，其被巨片“吞噬”的概率越大。

连通子图节点数递增趋势为何会在2013年打破还有待探究。

### 度分布数据分析

以上探究了协作关系网络的宏观特征，本小节着重分析节点度的分布特征，各年度范围内节点度的分布如图5-10所示。

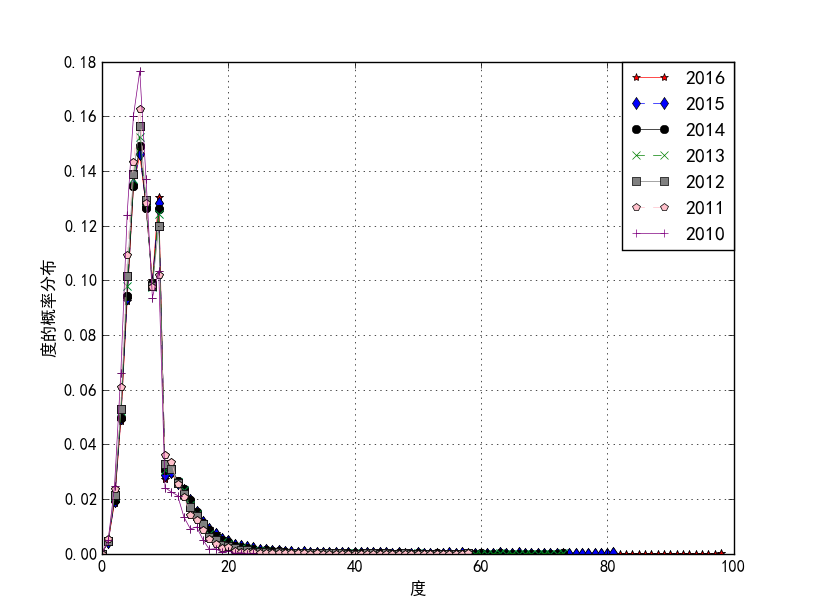


图5-10历年协作关系网络节点度分布

可以看出各年度节点的度大部分都分布在20以内，同时各年度范围内度的分布曲线重合度较好，曲线最高点有向下偏移的趋势，同时曲线尾部有外扩的趋势，见局部图5-11和图5-12。

各年度协作关系网络大部分节点的度分布在20以内，具有较大的度的节点很少，这和统计学现象非常吻合。节点具有较大度在一定程度上可以表明该作者参加项目次数较多，而具有度较大的节点数目极少，这种现象表明项目立项率极高的人极少，这一类人在历年项目立项和项目协作中具有极为重要的地位，极有可能是该领域的权威学者或有重要影响力的学者。通过分析节点的度、节点总权重和总权重对度的平均等因素，结合核心作者和挂名作者在协作关系网络中的特征，建立相应模型可以进行寻找核心作者和挂名作者等应用。

随着年份越接近当前年份，曲线最高点有向下偏移的趋势，同时曲线尾部有外扩的趋势，这表明节点平均度有增大的趋势。随着数据的积累，新加入的节点存在于原来网络中的概率变大，也就是小连通子图跟巨片之间存在媒介的概率变大。在连通子图被巨片“吞噬”的同时媒介的度也在增加，所以随着时间的推移，节点的平均度有增大的趋势，只是这个变化趋势越不明显。

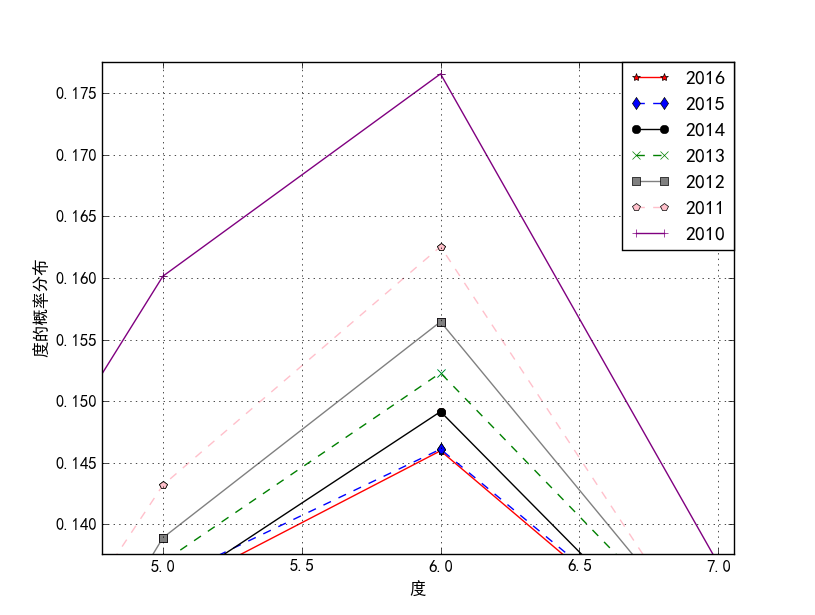


图5-11 曲线最高点局域图

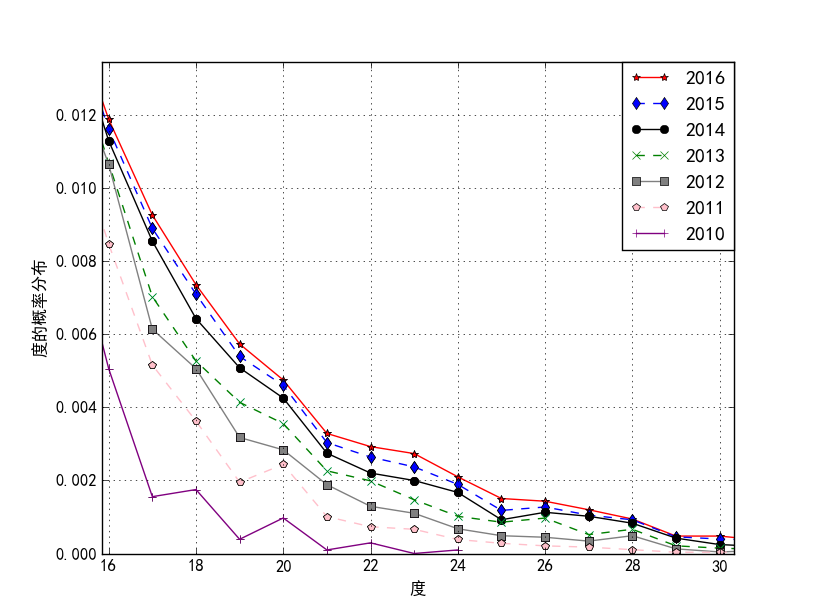


图5-12 曲线尾部局域图

通过对协作关系网络的分析，我们得到各年度范围内节点平均度如表5.5所示：

表5.5各年度范围内节点平均度

|  |  |  |
| --- | --- | --- |
| 年度 | 节点平均度 | 增量 |
| 2010-2010 | 6.61 | - |
| 2010-2011 | 7.11 | 7.56% |
| 2010-2012 | 7.38 | 3.79% |
| 2010-2013 | 7.53 | 2.03% |
| 2010-2014 | 7.71 | 2.39% |
| 2010-2015 | 7.78 | 0.91% |
| 2010-2016 | 7.85 | 0.89% |

可以看出随着时间的推移，节点平均度呈增加的趋势，但时间越接近当前年份，其平均度增量越小，即增长趋势越不明显。因为随着数据的积累，节点数越来越多，充当媒介节点所引起度的增长被大量新增的节点和原有节点所稀释。图5-12中各分布曲线向外扩的趋势逐渐减弱以及各曲线具有较好的重合度也证明了节点平均度的变化趋势越发不明显的特征。可以预见的是：随着数据的积累，曲线外扩程度会越来越小，尾部会逐渐趋于重合。

## 本章小结

本章主要对推荐系统的基本功能进行了测试，并对测试结果和推荐质量进行分析，从对测试结果和推荐质量分析上可知，推荐系统达到了设计要求和预期效果，满足了为科研合作者推荐潜在合作者的服务需求。同时本章尝试对协作关系网络呈现的一系列现象和规律进行探索和分析，为进一步研究协作关系网络夯实基础。

# 总结与展望

## 论文总结

本文是对提高科研管理数据资源利用率和解决广大科研工作者寻求合作者效率和质量低下问题的一个初步尝试，试图通过先进的计算机技术和理论，实现对科研管理数据资源的利用，并在一定程度上提高广大科研人员寻找合作者效率和质量。促进科研人员之间的交流，一定程度上可以起到增强科研成果和协作质量的效果。以下是本文的主要内容：

1）确定了课题需要解决的问题。通过对科研管理研究数据现状和广大科研工作者在寻求高质量合作者过程中遇到的问题进行分析，确定了本课题要解决的实际问题，并对该问题进行了拆分。

2）对相关理论、技术及架构的学习和选型。通过对传统推荐系统的相关理论和技术的学习，并结合本课题所要解决问题的实际情况，确定了本推荐系统所采用的理论原型——基于关系网络的推荐。调研并学习了基于图的相关理论、基于SSH的Web开发框架技术、基于面向架构的Thrift通信技术、基于Python数据处理的Networkx框架、基于前端可视化的Highcharts和D3技术等。

3）系统设计和实现。通过分析传统推荐系统原型在解决本课题实际问题时表现的不足，设计并提出了一套适用于该课题的推荐模型。包括数据清洗的设计、协作网络模型的设计、协作关系相似度和研究领域相似度的设计等。同时基于前期对相关理论和技术的调研、学习和研究，实现了推荐模型的基础框架，并在此框架上对外提供推荐服务和相关的数据分析。

4）系统测试及数据分析。对推荐系统进行了测试并对结果进行了简要的分析，同时针对在系统设计与实现过程中协作关系网络呈现的一些现象和规律做了简要的探究和分析。

综上所述，本文从系统的需求分析、技术及架构选型、系统设计、编码实现、系统测试以及协作关系网络探究等方面介绍了基于项目协作关系的合作者推荐系统的设计与实现过程。该系统借助于挖掘科研管理研究数据中的项目及人员信息，提高了数据的利用率，同时为科研工作者快速高效的寻找合作者提供了服务，能够促进科研人员之间的有效合作和有效交流，进而提高协作质量，在一定程度上为促进学科发展和增强科研质量提供了新的思路和方向，进一步推进了高校信息化平台的建设。

## 论文展望

到目前为止，本课题的研究已经告一阶段，通过基于项目协作关系的合作者推荐系统可以实现对给定作者推荐其潜在合作者，但是由于本人知识水平和时间等因素的限制，仍然还存在一些需要完善和值得探索的地方，现简述如下：

1. 用户反馈收集

目前实现的合作者推荐系统不包含用户对推荐结果的反馈功能。一方面可以通过分析用户对推荐结果的反馈来表征推荐系统的推荐质量；另一方面可以通过分析用户对推荐结果的反馈来完善推荐模型和调节参数，使得推荐系统更好的为用户服务。下一步可以考虑加入收集用户推荐结果反馈和反馈结果分析功能，为进一步优化模型和提高推荐质量服务。

1. 协作关系网络探究

目前仅对协作关系网络进行了简要的探究和应用，后续通过深入学习网络科学理论，可以进一步分析协作关系网络的生长、演变状况以及群组的生成发展机制；通过对网络社团结构的分析和检测，可以分析网络社团的构成情况；通过分析节点度的分布、节点强度等因素，可以进一步推荐核心作者和检测挂名作者等。除了提及到的探索方向之外，还有众多的未知的方向值得探索，但是以上任何一个方向都具有进一步分析和探究的意义。

1. 应用的深化

目前合作者推荐系统仅仅只是对给定的作者进行潜在合作者推荐，我们可以深化应用，通过推荐系统把该平台打造成科研合作者的交友平台，使得科研人员之间除了项目协作交流之外还可以进行在线交流，极大的促进了科研人员之间的交流，有助于新的科研成果、科研创新和研究方向的产生。

# 致 谢

弹指一挥间，时光如白驹过隙，转眼间充实而又忙碌的三年研究生生涯即将结束。在这三年的学习生活中，有过喜悦、也有过悲伤、有过成功、也有过挫折、有过付出、也有过回报。研究生期间不仅积累了专业知识，还学会了对人生的思考，不仅开拓了视野，还学会了为人处世，这些都将成为宝贵的财富，陪伴着我。在此，我衷心的感谢这三年来默默支持、关心和鼓励我的人。

首先，要感谢我的导师王玉明副教授，本论文在是王老师悉心指导下完成的。王老师治学严谨、待人真诚、学识渊博，不仅在学习上给予指导，在生活中也给予我莫大帮助和支持。王老师教会了我如何思考，如何做事，如何沟通，这些让我终生受益。在此，向王老师表示由衷的感谢。

感谢曾经帮助过我的老师和同学，在我最迷茫的时候，有你们的支持和鼓励，使得我能够不断前行。感谢实验室和宿舍的小伙伴们，他们是冯哲奇、彭亮、张楠、刘甲、林晓佳、冯子健、黄骏、姚雨涛、欧霄、钟远维、汪贝贝、黄钊伟、涂云、黄山、佘健夫等。你们在生活上给予我莫大的包容，在学习上给予我很大的帮助和支持，是你们使我这三年的研究生生活丰富多彩，祝愿大家在以后的日子一帆风顺。

感谢一直支持我的朋友们，他们是汤显才、熊兵兵、刘浩、于国星、张弛、徐江伟、张陶盛、张毛等，虽有大家身处天南地北，但是在我遇到困难和挫折的时候，你们的鼓励和支持让我有了不断前行的动力。

最后，要感谢我的家人，你们给了我最为无私的爱和无条件的支持。感谢你们在二十多年里对我的关怀、照顾和包容，亲情是至高无上的，值得我一辈子珍惜，因为你们，我才有更大的勇气和信心不断前行。希望你们身体健健康康，快乐幸福每一天。

# 参考文献

1. Grishman, Ralph. “Information Extraction.” IEEE Intelligent Systems 30 (2015): 8-15.
2. Cui, Lei et al. “Neural Open Information Extraction.” ACL (2018).
3. Hass, Richard W.. “Semantic search during divergent thinking.” Cognition 166 (2017): 344-357.
4. Hoffmann, Michael et al. “Benchmarking question answering systems.” Semantic Web 10 (2019): 293-304.
5. Shi, Baoxu and Tim Weninger. “Open-World Knowledge Graph Completion.” AAAI (2018).
6. Chinchor, Nancy and Elaine Marsh. “Appendix D: MUC-7 Information Extraction Task Definition (version 5.1).” MUC (1998).
7. Doddington, George R. et al. “The Automatic Content Extraction (ACE) Program - Tasks, Data, and Evaluation.” LREC (2004).
8. Gábor, Kata et al. “SemEval-2018 Task 7: Semantic Relation Extraction and Classification in Scientific Papers.” SemEval@NAACL-HLT (2018).
9. Verhagen, Marc et al. “SemEval-2007 Task 15: TempEval Temporal Relation Identification.” SemEval@ACL (2007).
10. Hendrickx, Iris et al. “SemEval-2010 Task 8: Multi-Way Classification of Semantic Relations between Pairs of Nominals.” SemEval@ACL (2010).
11. Miller, Scott et al. “A Novel Use of Statistical Parsing to Extract Information from Text.” ANLP (2000).
12. Kambhatla, Nanda. “Combining Lexical, Syntactic, and Semantic Features with Maximum Entropy Models for Information Extraction.” ACL (2004).
13. Moncecchi, Guillermo et al. “A survey of kernel methods for relation extraction.” (2010).
14. Zhao, Shubin and Ralph Grishman. “Extracting Relations with Integrated Information Using Kernel Methods.” ACL (2005).
15. Bunescu, Razvan C. and Raymond J. Mooney. “Subsequence Kernels for Relation Extraction.” NIPS (2005).
16. Culotta, Aron et al. “Integrating Probabilistic Extraction Models and Data Mining to Discover Relations and Patterns in Text.” HLT-NAACL (2006).
17. Mintz, Mike et al. “Distant supervision for relation extraction without labeled data.” ACL/IJCNLP (2009).
18. Min, Bonan et al. “Distant Supervision for Relation Extraction with an Incomplete Knowledge Base.” HLT-NAACL (2013).
19. Angeli, Gabor et al. “Combining Distant and Partial Supervision for Relation Extraction.” EMNLP (2014).
20. Färber, Michael et al. “Linked data quality of DBpedia, Freebase, OpenCyc, Wikidata, and YAGO.” Semantic Web 9 (2018): 77-129.
21. Takamatsu, Shingo et al. “Reducing Wrong Labels in Distant Supervision for Relation Extraction.” ACL (2012).
22. Yao, Limin et al. “Collective Cross-Document Relation Extraction Without Labelled Data.” EMNLP (2010).
23. Riedel, Sebastian et al. “Modeling Relations and Their Mentions without Labeled Text.” ECML/PKDD (2010).
24. Hoffmann, Raphael et al. “Knowledge-Based Weak Supervision for Information Extraction of Overlapping Relations.” ACL (2011).
25. Surdeanu, Mihai et al. “Multi-instance Multi-label Learning for Relation Extraction.” EMNLP-CoNLL (2012).
26. Min, Bonan et al. “Distant Supervision for Relation Extraction with an Incomplete Knowledge Base.” HLT-NAACL (2013).
27. Angeli, Gabor et al. “Combining Distant and Partial Supervision for Relation Extraction.” EMNLP (2014).
28. Goldwater, Sharon. “Part of Speech Tagging.” Encyclopedia of Machine Learning and Data Mining (2017).
29. Tan, Zhixing et al. “Deep Semantic Role Labeling With Self-Attention.” AAAI (2018).
30. Collobert, Ronan et al. “Natural Language Processing (almost) from Scratch.” Journal of Machine Learning Research 12 (2011): 2493-2537.
31. LeCun, Yann et al. “Deep Learning.” Nature 521 (2015): 436-444.
32. Socher, Richard et al. “Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank.” EMNLP (2013).
33. Zeng, Daojian et al. “Relation Classification via Convolutional Deep Neural Network.” COLING (2014).
34. Nguyen, Thien Huu and Ralph Grishman. “Relation Extraction: Perspective from Convolutional Neural Networks.” VS@HLT-NAACL (2015).
35. Xu, Kun et al. “Semantic Relation Classification via Convolutional Neural Networks with Simple Negative Sampling.” EMNLP (2015).
36. Santos, Cícero Nogueira dos et al. “Classifying Relations by Ranking with Convolutional Neural Networks.” ACL (2015).
37. Zeng, Daojian et al. “Distant Supervision for Relation Extraction via Piecewise Convolutional Neural Networks.” EMNLP (2015).
38. Miwa, Makoto and Mohit Bansal. “End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures.” CoRR abs/1601.00770 (2016): n. pag.
39. Yao, Kaisheng et al. “Depth-Gated LSTM.” CoRR abs/1508.03790 (2015): n. pag.
40. Peng, Nanyun et al. “Cross-Sentence N-ary Relation Extraction with Graph LSTMs.” Transactions of the Association for Computational Linguistics 5 (2017): 101-115.
41. Song, Linfeng et al. “N-ary Relation Extraction using Graph-State LSTM.” EMNLP (2018).
42. Lin, Yankai et al. “Neural Relation Extraction with Selective Attention over Instances.” ACL (2016).
43. Wang, Linlin et al. “Relation Classification via Multi-Level Attention CNNs.” ACL (2016).
44. Du, Jinhua et al. “Multi-Level Structured Self-Attentions for Distantly Supervised Relation Extraction.” EMNLP (2018).
45. Wu, Yi et al. “Adversarial Training for Relation Extraction.” EMNLP (2017).
46. Qin, Pengda et al. “DSGAN: Generative Adversarial Training for Distant Supervision Relation Extraction.” ACL (2018).
47. Vo, Duc-Thuan and Ebrahim Bagheri. “Self-training on refined clause patterns for relation extraction.” Inf. Process. Manage. 54 (2018): 686-706.
48. Su, Yu et al. “Global Relation Embedding for Relation Extraction.” NAACL-HLT (2018).
49. Zhang, Qianqian et al. “A Review on Entity Relation Extraction.” 2017 Second International Conference on Mechanical, Control and Computer Engineering (ICMCCE) (2017): 178-183.
50. Kumar, Shantanu. “A Survey of Deep Learning Methods for Relation Extraction.” CoRR abs/1705.03645 (2017): n. pag.
51. Mikolov, Tomas et al. “Distributed Representations of Words and Phrases and their Compositionality.” NIPS (2013).
52. Leimeister, Matthias and Benjamin J. Wilson. “Skip-gram word embeddings in hyperbolic space.” CoRR abs/1809.01498 (2018): n. pag.
53. Shi, Weiwei and Sheng Gao. “Relation Extraction via Position-Enhanced Convolutional Neural Network.” 2017 International Conference on Intelligent Environments (IE) (2017): 142-148.
54. Wang, Jianyong et al. “Recurrent Neural Networks With Auxiliary Memory Units.” IEEE Transactions on Neural Networks and Learning Systems 29 (2018): 1652-1661.
55. He, Kaiming et al. “Deep Residual Learning for Image Recognition.” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016): 770-778.
56. Röder, Michael et al. “GERBIL - Benchmarking Named Entity Recognition and Linking consistently.” Semantic Web 9 (2018): 605-625.
57. “Software - The Stanford Natural Language Processing Group.” Stanford.Edu, 1 Jan. 2014, nlp.stanford.edu/software/.
58. “TensorFlow.” TensorFlow, 1 Jan. 2019, www.tensorflow.org/.
59. “Neo4j Graph Database Platform.” Neo4j Graph Database Platform, 19 Oct. 2017, neo4j.com/.
60. Bostock, Mike. “D3.Js - Data-Driven Documents.” D3js.Org, 1 Jan. 2013, d3js.org/.
61. Ding, Boyang et al. “Leveraging Text and Knowledge Bases for Triple Scoring: An Ensemble Approach - The BOKCHOY Triple Scorer at WSDM Cup 2017.” CoRR abs/1712.08356 (2017): n. pag.
62. Wikipedia Contributors. “Knowledge Base.” Wikipedia, Wikimedia Foundation, 17 Feb. 2019, en.wikipedia.org/wiki/Knowledge\_base.
63. Zhang, Qi et al. “Adaptive Co-attention Network for Named Entity Recognition in Tweets.” AAAI (2018).
64. Sil, Avirup et al. “Neural Cross-Lingual Entity Linking.” AAAI (2018).
65. Goodfellow, Ian et al. “Deep Learning.” MITPress, 2016, www.deeplearningbook.org.
66. Ranganathan, Varun and S. Natarajan. “A New Backpropagation Algorithm without Gradient Descent.” CoRR abs/1802.00027 (2018): n. pag.
67. Gruslys, Audrunas et al. “Memory-Efficient Backpropagation Through Time.” NIPS (2016).
68. Pascanu, Razvan et al. “On the difficulty of training recurrent neural networks.” ICML (2013).
69. Dey, Rahul and Fathi M. Salem. “Gate-variants of Gated Recurrent Unit (GRU) neural networks.” 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS) (2017): 1597-1600.
70. Kanai, Sekitoshi et al. “Preventing Gradient Explosions in Gated Recurrent Units.” NIPS (2017).
71. Russakovsky, Olga et al. “ImageNet Large Scale Visual Recognition Challenge.” International Journal of Computer Vision 115 (2015): 211-252.
72. Krizhevsky, Alex et al. “ImageNet Classification with Deep Convolutional Neural Networks.” Commun. ACM 60 (2012): 84-90.
73. Agarap, Abien Fred. “Deep Learning using Rectified Linear Units (ReLU).” CoRR abs/1803.08375 (2018): n. pag.
74. Abadi, Martín et al. “TensorFlow: A System for Large-Scale Machine Learning.” OSDI (2016).
75. Rohekar, Raanan Y. Yehezkel et al. “Bayesian Structure Learning by Recursive Bootstrap.” NIPS (2018).

1. <http://image-net.org/challenges/LSVRC/2015/> and <http://cocodataset.org/#detections-challenge2015> [↑](#footnote-ref-1)
2. [↑](#footnote-ref-2)