|  |
| --- |
| 电子科技大学  UNIVERSITY OF ELECTRONIC SCIENCE AND TECHNOLOGY OF CHINA  硕士学位论文  MASTER THESIS  C:\Documents and Settings\Administrator\桌面\科大图标.jpg |

论文题目 社交网络数据采集方法研究及系统实现

学科专业 通信与信息系统

学 号 201521010811

作者姓名 杨杰

指导教师 钱峰 副教授

分类号 密级

UDC注1

学 位 论 文

社交网络数据采集方法研究及系统实现

（题名和副题名）

杨杰

（作者姓名）

指导教师 钱峰 副教授

电子科技大学 成 都

（姓名、职称、单位名称）

申请学位级别 硕士 学科专业 通信与信息系统

提交论文日期 论文答辩日期

学位授予单位和日期 电子科技大学 2018年6月

答辩委员会主席

评阅人

注1：注明《国际十进分类法UDC》的类号。

|  |
| --- |
| **Program State Monitoring and Error Recovery In Private Cloud Environment** |

A Master Thesis Submitted to

University of Electronic Science and Technology of China

|  |  |
| --- | --- |
| Major: | **Communication and Information Systems** |
| Author: | **Jie Yang** |
| Supervisor: | **Associate Prof. Feng Qian** |
| School: | **School of Communication & Information** |
|  | **Engineering** |

**独创性声明**

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，也不包含为获得电子科技大学或其它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。

作者签名： 日期： 年 月 日

**论文使用授权**

本学位论文作者完全了解电子科技大学有关保留、使用学位论文的规定，有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许论文被查阅和借阅。本人授权电子科技大学可以将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、汇编学位论文。

（保密的学位论文在解密后应遵守此规定）

作者签名： 导师签名：

日期： 年 月 日

# 摘 要

进入互联网时代以来，随着计算机技术和网络技术的发展，越来越多的应用场景催生出海量的数据处理任务。云计算将大规模的计算资源通进行互联和整合，实现了一种弹性的服务模式，使得计算资源（计算、存储、网络）能够像公共基础设施（水、电）一样提供给用户，是大规模数据处理问题良好的解决方案。同时，软件技术的飞速发展使得计算机程序应用范围越来越广，功能越来越复杂，也使得程序产生错误的可能性大大提高。如何有效地发现程序的异常和错误并加以处理在实际工作中有着非常重要的意义。

社交网络信息在用户行为研究和突发性事件分析领域有着重要的价值，然而社交网络信息的采集程序通常面临采集数据量大、运行时间长等问题。本文深入研究了云计算技术、程序状态监控和错误恢复相关理论，详细分析了采集程序的特点，设计并实现了一套私有云环境下的社交网络信息采集系统，并且实现了采集程序的状态监控和错误恢复。本文的主要工作包括：

1. 设计并实现了私有云环境下的社交网络信息采集系统。该系统提供通信服务、作业的执行与监控、作业的分派与调度、以及用户交互等主要功能。借助于私有云环境强大的计算能力，对采集作业进行分布式并行化处理，从而提高了采集作业的执行效率。

2. 实现了采集程序的状态监控、异常检测以及错误恢复。本文通过获取采集程序运行时的状态数据（系统资源使用情况和系统调用序列），并且将状态数据输入朴素贝叶斯分类器，实现了采集程序的状态监控和异常检测；同时，本文分析了采集程序的特点，结合应用级检查点技术，对于采集程序中关键的应用级数据进行检查点设置，实现了采集程序的错误恢复。

3. 对采集系统的通信服务功能、作业执行控制功能以及采集程序的异常检测、错误恢复机制进行了一系列测试。测试结果验证了采集系统具有足够的通信服务能力，验证了采集系统作业执行控制功能、采集程序异常检测机制以及错误恢复机制的正确性。

**关键词：**云计算，状态监控，错误恢复，检查点，信息采集系统

# ABSTRACT

In the Internet era, with the development of computer and network technology, numerous massive data processing tasks have been generated by many application scenarios. Cloud computing connects and integrates large-scale computing resources，thus provides elastic service. This kind of service provides computing resources (computation, storage, network) as public utilities (water, electricity) to users. Cloud computing is a sound solution to massive data processing problems. Meanwhile, rapid development of software technology results in more extensive usage and more complex functionality of computer programs, and also causes the probability of program faults to increase. It is of great significance that program faults should be identified and well-handled in practical work.

Social network information is very valuable in the fields of user behavior research and emergent events analyzing. However, using computer program to collect social network information usually faces the problems such as enormous amount of data and extremely long execution time. This thesis thoroughly studies cloud computing technology along with program state monitoring and error recovery theory, analyzes the characteristics of data-collecting programs in detail, then designs and implements a social network information collecting system in private cloud environment. This thesis contains the following main aspects of work:

1. Designs and implements a social network information collecting system in private cloud environment. The system has the following main functionality: communication service, job execution and monitoring, job dispatching and scheduling, and user interaction. Using the strong computing power of private cloud environment, the system is able to process jobs in a distributed and parallel manner, thus increases the efficiency of job execution.

2. Implements state monitoring, anomaly detection, and error recovery mechanism for information collecting programs. This thesis snapshots information collecting program during execution to acquire its state data (system resource usage data and system call sequence), then inputs state data into a Naïve Bayesian Classifier to implement state monitoring and anomaly detection. Also, this thesis analyzes the characteristics of information collecting programs, then uses application level checkpointing technique to checkpoint key application data, thus implements error recovery mechanism.

3. Runs a series of tests on the communication service and job execution control functionality of the information collecting system, and also runs a series of tests on the anomaly detection and error recovery mechanism of information collecting programs. The results of the above tests verify that the information collecting system has sufficient capability of communication service, verify the correctness of the information collecting system’s job execution control functionality, and also verify the correctness of anomaly detection and error recovery mechanism of information collecting programs.

**Keywords:** cloud computing, state monitoring, error recovery, checkpointing, information collecting system

# 目 录

[第一章 绪论 1](#_Toc483301079)

[1.1 研究背景及意义 1](#_Toc483301080)

[1.2 国内外研究现状 2](#_Toc483301081)

[1.2.1 社交网络数据采集方法 2](#_Toc483301082)

[1.2.2 社交网络数据查询方法 4](#_Toc483301083)

[1.3 本文主要工作与创新点 6](#_Toc483301084)

[1.4 论文结构安排 6](#_Toc483301085)

[第二章社交网络数据采集与查询系统设计 7](#_Toc483301086)

[2.1 云计算相关研究 7](#_Toc483301087)

[2.1.1 云计算的服务模型 7](#_Toc483301088)

[2.1.2 云计算的部署模型 8](#_Toc483301089)

[2.1.3 云计算的关键技术 9](#_Toc483301090)

[2.2 程序状态监控和错误恢复相关研究 13](#_Toc483301091)

[2.2.1 动态程序分析 13](#_Toc483301092)

[2.2.2 基于系统调用序列的程序异常检测 15](#_Toc483301093)

[2.2.3 检查点和回滚恢复 17](#_Toc483301094)

[2.3 本章小结 19](#_Toc483301095)

[第三章 私有云环境社交网络信息采集系统的设计与实现 20](#_Toc483301096)

[3.1 背景 20](#_Toc483301097)

[3.2 系统设计与实现 21](#_Toc483301098)

[3.2.1 系统总体架构 21](#_Toc483301099)

[3.2.2 消息通信库 22](#_Toc483301100)

[3.2.3 消息服务器 27](#_Toc483301101)

[3.2.4 计算节点守护进程 30](#_Toc483301102)

[3.2.5 作业服务器 32](#_Toc483301103)

[3.2.6 客户端 36](#_Toc483301104)

[3.3 系统测试 38](#_Toc483301105)

[3.3.1 测试环境 38](#_Toc483301106)

[3.3.2 消息服务性能测试 39](#_Toc483301107)

[3.3.3 作业控制性测试 42](#_Toc483301108)

[3.4 本章小结 44](#_Toc483301109)

[第四章 社交网络信息采集程序状态监控和错误恢复 45](#_Toc483301110)

[4.1 背景 45](#_Toc483301111)

[4.2 采集程序状态监控 45](#_Toc483301112)

[4.2.1 状态数据采集 45](#_Toc483301113)

[4.2.2 朴素贝叶斯分类方法 56](#_Toc483301114)

[4.2.3 测试方法与测试效果 58](#_Toc483301115)

[4.3 采集程序错误恢复 59](#_Toc483301116)

[4.3.1 应用级检查点的实现 59](#_Toc483301117)

[4.3.2 测试方法与测试效果 63](#_Toc483301118)

[4.4 本章小结 64](#_Toc483301119)

[第五章 结束语 65](#_Toc483301120)

[5.1 工作总结 65](#_Toc483301121)

[5.2 工作展望 66](#_Toc483301122)

[致 谢 67](#_Toc483301123)

[参考文献 68](#_Toc483301124)

[攻硕期间取得的研究成果 71](#_Toc483301125)

# 图目录

[图1-1 云计算产业规模现状及其预测情况 3](#_Toc483233082)

[图2-1 云计算服务模型层次关系以及各个层次的典型服务 7](#_Toc483233084)

[图2-2 GFS系统架构 10](#_Toc483233085)

[图2-3 BigTable的存储结构 11](#_Toc483233086)

[图2-4 MapReduce的处理框架 12](#_Toc483233087)

[图2-5 Pin软件架构图 14](#_Toc483233088)

[图2-6 Sekar方法根据示例代码建模得到的FSA 17](#_Toc483233089)

[图3-1 私有云环境社交网络信息采集系统总体架构图 21](#_Toc483233090)

[图3-2 Boost::Asio异步IO模式操作流程（用户发起IO操作） 23](#_Toc483233091)

[图3-3 Boost::Asio异步IO模式操作流程（IO操作完成，执行回调函数） 23](#_Toc483233092)

[图3-4 保证消息服务器进程实例唯一性的操作流程 29](#_Toc483233093)

[图3-5 守护进程后台化流程 31](#_Toc483233094)

[图3-6 Scheduler作业调度的基本框架 35](#_Toc483233095)

[图3-7 MainWindow界面示意图 37](#_Toc483233096)

[图3-8 JobMonitorWindow子界面示意图 38](#_Toc483233097)

[图3-9 消息服务器性能测试模型 39](#_Toc483233098)

[图3-10不同SessionAPI收发对数量下消息服务器的吞吐量 41](#_Toc483233099)

[图3-11 进程号等于4465的作业暂停前 42](#_Toc483233100)

[图3-12 进程号等于4465的作业暂停后 42](#_Toc483233101)

[图3-13 进程号等于4465的作业恢复前 43](#_Toc483233102)

[图3-14 进程号等于4465的作业恢复后 43](#_Toc483233103)

[图4-1 /proc/net/tcp文件的内容 53](#_Toc483306605)

[图4-2 /proc/[pid]/fd目录下对应socket描述符的fd列表（pid=2724） 53](#_Toc483306606)

[图4-3 统计进程和连接收发数据量流程图 54](#_Toc483306607)

[图4-4 采集作业的基本流程 60](#_Toc483306608)

[图4-5 采集作业检查点设置流程 61](#_Toc483306609)

[图4-6 采集作业检查点恢复流程 62](#_Toc483306610)

[图4-7 向执行过程中的采集程序发送键盘中断信号 63](#_Toc483306611)

[图4-8 采集程序回滚恢复 64](#_Toc483306612)

# 表目录

表2-1 云计算服务模型比较表 8

[表3-1 消息数据结构的主要字段及其说明 24](#_Toc483233208)

[表3-2 消息通信库的主要组件 25](#_Toc483233209)

[表3-3 消息服务器的主要组件 28](#_Toc483233210)

[表3-4 计算节点守护进程的主要组件 30](#_Toc483233211)

[表3-5 作业服务器的主要组件 33](#_Toc483233212)

[表3-6 作业服务器需要处理的消息类型 33](#_Toc483233213)

[表3-7 客户端的主要组件 37](#_Toc483233214)

[表3-8 不同SessionAPI收发对数量下消息服务器性能测试数据 40](#_Toc483233215)

[表3-9 不同SessionAPI收发对数量下消息服务器的吞吐量 40](#_Toc483233216)

[表4-1 /proc/[pid]/stat文件部分参数名称及其含义 46](#_Toc483233217)

[表4-2 /proc/[pid]/statm文件部分参数名称及其含义 47](#_Toc483233218)

[表4-3 /proc/[pid]/io文件部分参数名称及其含义 49](#_Toc483233219)

[表4-4 特征数据文件包含的特征项 58](#_Toc483233220)

[表4-5 观测数据的分类结果 59](#_Toc483233221)

# 第一章 绪论

## 1.1 研究背景和意义

社交网络(Social Network Service)源自于人们在网络中的交流与分享，最开始的网络交流来自于电子邮件，这种方式主要是点对点通信，接着BBS实现了群发与转发这两种功能，更进一步推进了社交网络的发展，将点到点交流变为了点到面交流，随后，即时通信(IM)和博客(Blog)扩展了BBS的功能，提高了信息传输速度以及分享各种各样的感受与知识。随着网络科技不断的发展，如今在互联网中网络社交已经融入了人们的日常生活，中国互联网络信息中心(CNNIC)在京发布第40次《中国互联网络发展状况统计报告》[1]，截至2017年6月，中国网民规模达到7.51亿，占全球网民总数的五分之一。2017年微博用户发展报告[2]中表明微博月活跃用户数量高达3.76亿。

社交网络数据挖掘的意义在于从海量的数据中发掘有用的信息与数据，并利用信息提高不同领域的效率。从社交网络诞生以来，不断有学者对其进行研究：汪全莉[1]等人将社交网络数据挖掘用于网络教育，通过网络数据挖掘，提取学生感兴趣的信息作为依据，从而提高了网络站点的效率；卢明泰[2]通过社交网络数据挖掘技术，收集了大量个人信息，并提出了个人标签云这个概念，为以后的研究个人状态与行为做好了铺垫；黄琴[3]将数据挖掘技术运用到了网络学习个性化推荐中，为个人定制个性需求，大大提高了学习效率。

如今的社交网络每时每刻都有海量的数据在其中诞生，同时又由于社交网络的不断发展以及用户需求不同，使得社交网络变得异常的复杂。社交网络的内容主要是以网页的形式出现的，大多数网页整体上都遵循W3C制定的DOM树型结构标准，而这一标准使得网页结构并不固定，每个社交网站都可以设计满足自己需求的独特风格。图1-1中(a)图是新浪微博某个用户的首页图，(b)图是某个Twitter用户的主页图。从图中我们可以看出这是两种完全不同风格的设计，用户的基本信息内容不同，内容展现的位置也不同。因此，对于传统的数据采集方法而言，从不同的社交网站获取内容需要针对每个网站设计特定的方法，这无疑增加了使用者难度，同时，从采集好的海量数据中查询与挖掘数据也存在不少难点。

因此，一个好的社交网络数据采集与查询系统拥有高效而简便的数据采集方法以及准确的数据查询方法，可以让使用者在海量的社交网络中快速获取完整的数据以及高效的从数据中查找相关信息，为用户节约大量时间。

图片包含 屏幕截图

已生成极高可信度的说明 

1. (b)

图1-1 某两个社交网络网站的网页图

## 1.2 国内外研究现状

写一点为什么需要研究下面两个

### 1.2.1 社交网络数据采集方法

数据采集【8】（https://en.wikipedia.org/wiki/Data\_collection）是以系统化方式收集信息的过程。数据采集是所有研究领域研究的一部分，包括自然科学，社会科学，人文科学和商业。虽然方法因研究领域而异，但确保准确和海量收集的重点仍然是一样的。所有数据采集的目标是获取高质量的数据，并通过分析数据可以得出令人信服的结论。

社交网络数据采集主要针对结构或半结构化的数据根据需求进行提取、储存，并将处理后的数据以网页形式呈现。自1997年HTML【9】（"HTML 4.0 Specification — W3C Recommendation — Conformance: requirements and recommendations". World Wide Web Consortium. December 18, 1997. Retrieved July 6, 2015.）正式版本发布以来，海量的社交网络数据就蕴藏在其中，现在常见的社交网络有新浪微博【10】、facebook[11]、twitter[12]、领英等。针对web网页信息提取的研究从社交网络诞生以来一直在进行。

1997年，J Hammer【11】（J Hammer Extracting semistructured information from the web Workshop on Management of Semistructured Data , 1997 :18--25）研究了一种使用python编写的web数据提取器，这种提取器需要根据用户输入采集数据的网页位置，确定采集路径，通过这种采集路径并再输入需要采集网页的链接，从而采集数据。这个方法有两个不足，一是抗风险能力低，一担网页结构有一点变化，那么必须重新确定采集路径，二是需要用户输入采集网页的链接，如果链接数量过多，那么会浪费大量的人工。

在2001年，V Crescenzi【12】（V Crescenzi ， G Mecca ， P Merialdo RoadRunner: Towards Automatic Data Extraction from Large Web Sites 2001）等3人通过研究web网页的html结构，比较两棵同源网页的DOM树之间的是否匹配，提出了一种包装器用于自动从网页中提取数据，这使得以前需要大量人力的数据提取，变得简单、方便。但是这种方式将DOM树每个节点的信息都采集了，其中包含了很多噪音，例如广告节点中的广告信息，网页本身的导航、尾部等，因此，这种方式采集的数据用户必须后期过滤处理才能使用。

紧接着第二年，在2002年时李效东与顾毓清【13】（李效东 ， 顾毓清 基于DOM的Web信息提取 《计算机学报》 , 2002 , 25 (5) :526-533）两人同样基于DOM树，将需要提取的数据在DOM树的位置看为一种坐标，并使用数学归纳法生成一条从根结点到目标节点路径的提取规则，这种提取方法同样有抗风险不足的特点。

2004年，周明建【14】（基于本体论的Web信息抽取 周明建 ， 高济 ， 李飞 《计算机辅助设计与图形学学报》 , 2004 , 16 (4) :535-541）等3人根据本体论，并以需要提取信息的层次结构作为提取路径，重新定义网络页面的信息本体,并自动解析生成网络页面的结构本体。通过对这两个本体进行对比，创建了一种算法来半自动地生成信息提取规则。这种方法主要用于网页主题提取，即通过网页本体，采集同一类型的网页，不过提取内容的准确性这点上有待提高。

2009年，李朝【15】（李朝 ， 彭宏 ， 叶苏南 ， 张欢 ， 杨亲遥 基于DOM树的可适应性Web信息抽取 《计算机科学》 , 2009 , 36 (7) :202-203）等人对网页提取规则自适应进行了研究，认为当网页结构改变后，先前生成的提取规则无法使用改变后的结构，因此提出了一种通过聚类方法获取关键词组，并利用DOM树结构确定关键词组的位置，从而达到自适应的目的。这种方式能够有效的抵抗网页改变这种风险，每当网页结构改变时，重新采集以前采集过的网页，获取所有文本内容，然后通过关键词，重新定位采集内容的位置，并生成采集路径，这使得网页信息提取技术的自适应进一步提高。

### 1.2.2 社交网络数据查询方法

社交网络数据查询是信息检索技术与社交网络数据相结合的查询技术。现代的信息检索开始于电脑发明之后，人们意识到电脑可以用于存储和机械检索大量的信息，在1945年Vannevar Bush发表了一篇名为“As We May Think”【1】（Vannevar Bush. As We May Think. Atlantic Monthly, 176:101–108, July 1945）的文章，从这篇文章中人类社会诞生了海量信息存储这个概念，由此，计算机与海量数据信息检索联系到了一起，并不断向前发展。

在1957年，Luhn提出用词来检索文档【17】（[17] H. P. Luhn. A statistical approach to mechanized encoding and searching of literary information. IBM Journal of Research and Development, 1957.）。他通过统计分析感兴趣的文档集合，建立一组概念以及表达这些概念的词汇表，同时建立同义词库类型的词典与索引，以后在搜索文档时，就可以通过词典判断文档类型。这篇文章为今后以词作为检索重要依据提供了思路。

在1983年，G Salton 与MJ Mcgill在《介绍现代信息检索》【3】（G Salton ， MJ Mcgill Introduction to Modern Information Retrieval 《Library Resources & Technical Services》 , 1983 , 55 (4) :239-240）中阐述了互联网发展带来的巨大信息量，以及对出版物的巨大影响。这表明了互联网开始取代传统的纸质媒体，人们开始更加倾向于从互联网中获取与交流信息，信息检索技术在互联网得到进一步发展。

随后，相继出现了布尔模型，概率模型，TFIDF等方法，但是这些信息检索可能会带来检索返回信息不准确，不全面等问题，例如，“murder”与“kill”这两个查询词都有表示杀人的意思，当用户输入“kill”想要查询一间杀人事件时，可能无法获取到相关信息，而相关反馈这种查询修改技术，通过对查询词的扩展，可以有效的查询得到用户想要的隐藏信息，改进了检索结果。在2000年，J Rocchio在《信息检索中的相关反馈》【4】（J Rocchio Relevance feedback in information retrieval 《Computer Science》 , 2000 :313-323）中通过制定一种扩展查询方法，表明了相关反馈可以提高传统的查询方法。

2005年，Finin【5】（T Finin ， J Mayfield ， A Joshi ， RS Cost ， C Fink Information Retrieval and the Semantic Web 2005）等5人在《信息检索与语义网页》中表明了语义也是信息检索技术的重要部分，文章指出拥有语义的网页内容可以作用于传统的检索系统，通过语义网页的推理结合传统的搜索可以改进或增强信息搜索的性能。 在2013年，Google开源了一款用于词向量计算的工具——word2vec，它的出现引起了业界广泛关注，该工具可以很好的度量词与词之间的相似性，为判断句子之间的相似性做出了巨大贡献。紧接着，Z Wang【6】（Z Wang ， H Mi ， A Ittycheriah Sentence Similarity Learning by Lexical Decomposition and Composition 2017）等3人在2017年发表了一篇名为《通过组合与分解判断句子的相似性》，这篇文章主要通过使用word2vec的原理，先通过比较词的相似性，然后比较句子语义的相同与不同部分，这种方式能够有效的提高短文本相似性判定，进一步表明语义在查询中起到了一定的作用。

## 1.3 本文主要工作与创新点

本文研究了社交网络的发展与现状，同时研究了社交网络数据采集方法与数据查询方法的发展史，根据日益增长的数据采集与数据查询需求与千变万化的社交网络数据矛盾的特点，设计了一套社交网络数据采集与查询系统，在这套系统中包含了独创的数据采集方法与数据查询方法，使得本文系统的适用性与实用性大大增强。

具体来说本文完成的主要工作包括如下几点：

1.设计了社交网络数据采集与查询系统。为了解决界面操作与数据展示、前后端交互、数据采集、数据查询和数据存储这5个功能需求，系统包含了3个功能模块——服务器端功能模块、客户端功能模块以及数据存储功能模块，其中系统的核心是服务器端的功能，而服务器端功能模块的核心是数据采集与数据查询方法，通过对数据采集与查询方法的分析分别提出了社交网络自动化数据采集方法与基于权重与语义的扩展查询方法。

2.提出并实现了社交网络自动化数据采集方法，此方法一共经历3个步骤，一是完成了重构DOM树，为接下来分析、提取、扩展做铺垫；二是完成了自适应数据采集代码的生成，为了增加采集网页的自适应，本文提出并实现了基于相对路径结合绝对路径的采集路径生成方法，这种方法增强了网页数据采集的自适应性；三是完成了同构网页链接扩展，通过比较网页相似性，获取扩展链接，接着生成链接扩展规则，实现快速扩展同构URL链接数量，同时经过扩展链接方法的测试分析，说明了扩展链接方法具有可扩展性、可靠性、唯一性以及随机性。

3.提出并实现了基于权重与语义的扩展查询方法，该方法在自动相关反馈的查询扩展方法上进行改进，在原本基于词频扩展的基础上引入了基于词汇分解与组合判断语义相似性的方法，同时基于局部上下文分析方法计算出了扩展查询词与原始查询词的不同权重，将这个权重融入原有查询模型。经过对数据的测试分析，结果表明了新的扩展查询方法提高了查询结果的准确率。

## 1.4 论文章节安排

本文由六个部分组成，具体章节安排如下：

第一章：介绍了本文的研究背景，以及社交网络数据采集方法与查询方法的发展历史；并且对本文的主要工作、创新点和文章结构进行说明。

第二章：提出需求功能，并根据需求设计了社交网络数据采集与查询系统，此系统包含了3大模块——服务器端、客户端、数据存储，同时提出了本文系统独创的数据采集与数据查询方法。

第三章：详细介绍了社交网络自动化数据采集方法，并分别说明了此方法实现的3个步骤——重构DOM树、自适应数据采集代码的生成以及同构网页链接扩展，并对网页链接扩展进行了相应测试。

第四章：详细介绍了基于权重与语义的扩展查询方法，包括方法思路以及具体实现，并对此方法进行测试与分析。

第五章：详细讲述了本文系统的测试环境，同时测试了系统社交网络数据采集与数据查询这两种功能，最后通过测试结果分别进行了分析。

第六章：全文工作总结以及将来的工作展望。

# 第二章 社交网络数据采集与查询系统设计

## 2.1 背景

人们在社交网络中的交流、分享等活动是在各式各样的社交网站中完成的，在国内较为出名的社交网站高达200多个，不同的社交网站涉及不同的领域，其中豆瓣着眼于各类生活爱好的分享，人人网专注于白领用户和学生用户的交流与娱乐，世纪佳缘将目标人群锁定为未婚男女，还有百度空间这样的多功能大众化社交网站。社交网站的数量逐年增加，传统的数据采集方法针对不同的网站需要专业人员设计不同的采集方法，这将造成人力资源的大量浪费，例如，一个用户需要获取领英成员的信息，他必须学习计算机相关数据采集知识或者聘请一名专业人员，针对领英网站的HTML结构，设计一种专门的数据采集路径，如果遇见网站改版(修改了HTML源码)的情况，又要重复进行之前的工作，而且当用户想要获取新浪微博的数据时，必须重新开始研究新浪的网页源码，耗时又耗力。同时拥有海量数据并不意味着拥有海量信息，只有从数据中提取出有用的信息，那么数据才是“活的”，所以信息的查询也同样关键。鉴于上述两种问题，本人研究了一种适应性较强的自动提取社交网络数据与查询数据的系统。社交网络数据采集与查询系统主要功能需求如下所示：

(1) 数据采集功能

数据采集功能是本文系统的核心功能之一。它必须能够从各式各样的网页中精确获取用户需要信息，并具备自适应网页结构小幅修改的情况，同时必须能够举一反三，根据用户提供的样本网页挖掘出所有同构网页，完善数据采集功能的扩展性，使数据变得完整。

(2) 数据查询功能

数据查询功能是系统的另一个核心功能。拥有海量的数据并不意味着拥有海量的信息，从数据中挖掘出用户关心的信息是数据查询功能的关键，用户在输入一组查询词词组后，数据查询功能必须不仅能够返回包含这些查询词的文本，同时能够扩展这些查询词，获取包含扩展词的文本信息，实现查询信息的完整性。

(3) 前后端交互功能

前后端交互功能应该具有以下能力：1)为前端的每个请求特制一个链接接口(API)；2)稳定接收前端的每一个请求；3)返回后台的数据给前端；4)保证前后端通信的稳定性，即能够在前后端交互中断时，重新自动连接。

(4) 数据存储功能

数据存储功能需要达到以下3点需求：1)海量数据存储；2)保证数据安全性、可扩展性；3)为查询数据文本建立高效索引，实现千万级数据秒数查询速度。

(5) 界面操作与数据展示功能

一个良好的系统必须要有对用户友好的操作界面，用户可以根据系统使用说明，快速使用系统各项功能。这个功能的意义在于1)便于用户操作；2)方便查看系统状态；3)展现数据采集进度；4)可视化数据，用户体验更加直观。

## 2.2 系统总体设计

（写的太少了，缺少分析）

社交网络数据采集与查询系统是一种集采集与查询一体的系统，包含有客户端、服务器端、数据存储3个部分，如图2-1所示。

客户端有3个模块，1)数据采集交互模块实现了采集数据可视化、采集数据路径确认；2)查询可视化模块实现了用户查询参数编辑、查询结果可视化；3)状态监控模块实现了数据采集监控、服务器资源监控、网络IO监控。

服务器端包含两个功能：数据采集功能与数据查询功能。数据采集功能先后经历3个阶段：DOM树构建、采集代码生成、链接扩展，数据查询功能包含查询词扩展与查询文本扩展两个步骤。

数据存储根据数据采集功能与数据查询功能的不同需求，分别对应Mongodb模块与ElasticSearch模块，Mongodb模块解决了数据采集功能海量数据安全、可靠的存储需求，ElasticSearch模块能够实现快速数据查询，提高了数据查询功能的效率。



图2-1 社交网络数据采集与查询系统总体架构图

## 2.3 系统功能

### 2.3.1 服务器端

服务器端是社交网络数据采集与查询系统的核心，首先服务器端需要接收客户端传送的各种请求，并将符合需求的数据返回给用户，然后服务器端需要稳定的API接口连接数据存储模块，实时与数据存储模块交互数据，最后服务器端需要实现数据采集功能与数据查询功能。因此，本人设计的社交网络数据采集与查询系统使用Python语言完成服务器端的构建。

Python[1], 是一种面向对象的解释型计算机程序设计语言，由荷兰人Guido van Rossum于1989年发明，第一个公开发行版发行于1991年。Python是纯粹的自由软件，源代码和解释器CPython遵循 GPL(GNU General Public License)协议。Python具有丰富和强大的库。它常被昵称为胶水语言，能够把用其他语言制作的各种模块（尤其是C/C++）很轻松地联结在一起。常见的一种应用情形是，使用Python快速生成程序的原型，然后对其中有特别要求的部分，用更合适的语言改写，比如3D游戏中的图形渲染模块，性能要求特别高，就可以用C/C++重写，而后封装为Python可以调用的扩展类库。



图2-2系统Django架构总览图

使用Python完成采集数据功能有两个好处，一是相比与其他静态编程语言，如java，C#，C++，Python抓取网页文档的接口更简洁，再者抓取的网页通常需要处理，比如过滤html标签，提取文本等，Python的BeautifulSoap提供了简洁的文档处理功能，能用极短的代码完成大部分文档的处理。同时Python有着大量的三方库，如用于自然语言处理的NTLK库，用于词频查询文本的GENSIM库，这为数据查询功能的实现提供了良好的基础。



图2-3 Django请求响应流程图

Django 是用Python开发的一个免费开源的Web框架，可以用于快速搭建高性能、优雅的系统，拥有强大的前后端交互能力。Django 拥有较强的数据库操作接口（QuerySet API），如需要也能执行原生SQL，便于服务器端与数据存储模块的数据交互，并且Django拥有强大、易扩展的模板系统，在处理前端请求时有着稳定的性能。

Django架构总览图如图2-2所示，Django包括HTTP请求处理/响应模块、URL映射模块、应用模块、对象关系映射模块、模板引擎模块、数据库模块与文件系统模块。其中，Django核心在于middleware（中间件），Django所有的请求、返回都由中间件来完成。中间件，就是处理HTTP的request和response的插件，比如有Request中间件、View中间件、Response中间件、Exception中间件等，Middleware都需要在 “project/settings.py” 中 MIDDLEWARE\_CLASSES 定义。大致的程序流程图如2-3所示，用户HTTP请求被Django接收，打包成请求对象类，如果需要直接返回Response响应，则进入Response中间件返回用户响应数据，否则进入View中间件，并判断是否出现异常，如果出现异常，那么进入Response中间件返回用户异常，否则调用在view文件中的url相应函数，并将需要返回的数据放入Response中间件，返回用户需要的数据。

服务器端最核心的两个功能是采集功能与查询功能。

采集功能必须实现对任意网页、任意数据的采集，因此采集功能首先需要重构DOM树，将结构化或半结构化的HTML代码变为采集功能能够识别与使用的标记，接着需要定位用户所需采集的数据所处DOM树的位置，生成定位路径代码，然后通过链接扩展扩展同一类的网页链接，最后采集全部链接的数据存入数据存储模块中。具体流程见第三章。



图2-3 Django请求响应流程图

查询功能需要做到将更多用户需要的文本查询出来，原始查询词是查询功能最重要的依据，但有一些文本可能没有包含这些查询词却是用户需要的文本，因此需要将查询词扩展，这些扩展词在接下来的查询过程中占有一定的比重，然后通过新的查询词再去查询用户所需的文本实现查询文本扩展，最后将查询到的结果返回给用户。具体流程见第四章。



图2-3 Django请求响应流程图

### 2.3.2 客户端

客户端使用超文本标记语言(HyperText Markup Language)进行构建，在网页中显示用户操作界面，客户端的构建使用Bootstrap框架。Bootstrap是一款由Twitter公司提供的前端框架，用于开发响应式布局、移动设备优先的 WEB 项目，同时Bootstrap 是一套用于 HTML、CSS 和 JS 开发的开源工具集，利用开发者提供的 Sass 变量和大量 mixin、响应式栅格系统、可扩展的预制组件、基于jQuery 的强大插件系统，能够快速为使用者开发出原型或者构建整个APP。

客户端包括3个界面，数据采集界面，数据查询界面，状态监控界面。



(a)



(b)

图2-4 数据采集界面（a）生成标签（b）采集数据

数据采集界面包含两个子界面，如图2-4所示。其中生成标签子界面用于用户确定采集内容与生成采集路径，采集数据子界面选定链接数据库与内容数据库后，有两种采集方式，一种是特定采集方式，这种方式仅仅采集用户给定的链接内容，另一种是扩展采集，主要用于扩展用户给定网页的同结构网页，使采集内容更加完整。

（数据查询界面 状态监控界面还没写）

### 2.3.3 数据存储

数据存储的目的在于将服务器端采集的数据永久、安全、有效的存储，同时提供高效的数据查询效率，因此数据存储功能引入MongoDB数据库与ElasticSearch搜索服务器。

MongoDB是一个基于分布式文件存储的数据库，是一个介于关系数据库和非关系数据库之间的产品，是非关系数据库当中功能最丰富，最像关系数据库的。支持的数据结构非常松散，因此可以存储比较复杂的数据类型。MongoDB最大的特点是支持的查询语言非常强大，其语法有点类似于面向对象的查询语言，几乎可以实现类似关系数据库单表查询的绝大部分功能，而且还支持对数据建立索引。在使用MognoDB存储海量数据时，搞数据量和吞吐量的数据库应用会对单机的性能造成较大的压力，在不增加更多的CPU与存储资源的情况下，使用水平扩展，将数据集分布在多个服务器上，这种分片方式能够实现存储海量数据时的高并发性与数据的安全性，MongoDB分片集群组件如3-1表所示。

表3-1 MongoDB分片集群组件

|  |  |
| --- | --- |
| 组件 | 说明 |
| Config Server | 存储集群所有节点、分片数据路由信息。 |
| Mongos | 提供对外应用访问，所有操作均通过mongos执行，一般有多个mongos节点，数据迁移和数据自动平衡。 |
| Mongod | 存储应用数据记录，一般有多个Mongod节点，达到数据分片目的。 |

图2-5展示了Mongodb架构，服务器端的应用（Application）连接MongoDB模块的Mongos，Mongos读取配置（Config）后，通过算法计算出数据存放点，将数据存入其中一个分片（Shard）中，每个分片是一个复制集，保证了数据的安全性。



图2-5 MongoDB架构

ElasticSearch是一个基于Lucene的搜索服务器，不但稳定、可靠、快速，而且也具有良好的水平扩展能力，是专门为分布式环境设计的。Elasticsearch可以进行4项工作：1）全文搜索功能；2）分布式实时文件存储，并将每一个字段都编入索引，使其可以被搜索；3）实时分析的分布式搜索引擎；4）可以扩展到上百台服务器，处理PB级别的结构化或非结构化数据。ElasticSearch搜索服务器用于数据查询，对于上亿数据查询仅需3-5秒，相对于MongoDB的查询效率提高了几十倍。（ElasticSearch架构图需不需要。。看情况）

## 2.4 系统采集与查询核心方法

### 2.4.1 数据采集方法

Web信息提取技术大体可以归类成4种【\*\*】（Web信息抽取技术综述 陈），基于统计理论的技术、基于视觉特征的技术、基于DOM树结构的技术、基于模板的技术。

基于统计理论的技术是通过统计给定网页的各个标签拥有的信息与普通文本拥有的信息比值以此获取网页的主题，这种技术不是研究具体的某个网页或者网站，而是对应不同的网站与网页。自从GUPTA等人【\*】在2003年提出正文区域普通文本占多数，而非正文部分一般是链接占多数以后，基于统计理论的信息提取技术不断发展，在2009年，周佳颖等人【\*】（周佳颖 ， 朱珍民 ， 高晓芳 基于统计与正文特征的中文网页正文抽取研究 《中文信息学报》 , 2009 , 23 (5) :80-85）利用正文特征克服了传统统计方法无法抽取多正文体网页的缺陷，实现抽取单正文和多正文的精确率分别为94%和91%。

基于视觉特征的信息提取技术是基于网页展现内容块位置的不同进行信息提取的。由于网页基本是由HTML编写的，而HTML是一种十分灵活的语言，这就造成了不同的网站甚至同一个网站不同内容的网页格式不一样，对于DOM树下同一个父节点的子节点展现在网页上的内容可能不是同一种类型，这不利于通过分析DOM树结构来确定网页主题信息，因此Cai等人【\*】在2003年提出了VIPS（vision-based page segmentation）算法用于视觉特征提取网页信息，该算法主要通过<hr><p><ul><table>等标签将网页切分成一个一个的块，并将内部节点大部分是文本的看作最小块，不再分割。

基于DOM树结构的信息提取技术着眼于DOM树，网页的HTML页面经过删除、补全等操作后可以形成一棵DOM树，DOM树每个节点有着自身特有的信息，如标签、节点文本等，通过分析节点的特性可以获取信息路径，这种方式对于基于统计与基于视觉特征的方式更加的简单与精确，因此获得了快速发展，如RoadRunner【V Crescenzi ， G Mecca ， P Merialdo RoadRunner: Towards Automatic Data Extraction from Large Web Sites Proc of the 27th VLDB Conference.San Francisco: Morgan Kaufmann Publishers,2001:109-118.】系统主要针对相同结构的网页进行数据提取，在推导出模板后，利用模板生成模板代码并对其他的网页进行信息提取。

基于模板的信息提取技术是对互联网中有着海量的只需往模板中填充数据就生成的网页的信息提取。张彦超等人【张彦超 ， 刘云 ， 李勇 ， 沈波 基于自动生成模板的Web信息抽取技术 《北京交通大学学报》 , 2009 , 33 (5) :40-45】提出了基于自动生成模板的Web信息抽取方法，可以消除网页噪声，精确获取数据，同时他们还提出了URL模板匹配的概念，即通过待信息提取的URL与URL模板进行匹配判断当前URL是否是所需采集的URL链接，这种方式快速扩展了同源同构网页。

对于上述4种网页信息提取技术，基于统计理论的技术主要针对网页主题、网页正文提取，对于具体网页上每一块的数据提取效果并不理想，而基于视觉特征的技术操作较为复杂，对于各种动态网页与多样性结构网页的信息提取不能很好适应，基于DOM树结构的信息提取技术能够准确提取指定数据，适应性优于视觉特征提取技术，同时基于模板的信息提取技术在对URL链接扩展方面有着自身的优势，因此本文使用DOM树提取所需数据并使用模板技术扩展链接，有利于数据准确提取与数据的完整性，可以自适应的获取同源同构的所有网页数据。

### 2.3.2 数据查询方法

数据查询模型有布尔模型、向量空间模型、概率模型、统计语言模型、神经网络模型等，对于查询方法中的扩展查询方法有基于用户反馈的查询扩展、基于局部分析的查询扩展与基于全局分析的查询扩展，其中最值得研究的是基于局部分析中的基于自动相关反馈的查询扩展(A Query Expansion Algorithm Based on Automatic Relevance Feedback)方法。

基于自动相关反馈的查询扩展方法是在相关反馈的基础上进行提高，这种方式不需要用户的参与，它的核心是：首先认定第一次查询的前N个文本是有效的，接着利用这前N个文本扩展出新的查询词，并与原始的查询词构成新的查询词集合，最后使用新的查询词集合再去查询文本集，获取相关文本成为最终查询结果，基于自动相关反馈的查询扩展流程如图1-1所示。



1-1 自动相关反馈的查询扩展流程图

基于自动相关反馈的查询扩展方法的具体描述如下：

(1)用户输入初始查询词集合Q以及查询文本集合D；

(2)将Q与D转化为相关向量与，其中，；

(3)使用BM25方法查找原始文本集，计算出每个文本与查询词集Q的相似性并进行排序，取出前N个文本，作为扩展文本集；

(4)通过某种方式计算出扩展文本集每个词的权重；

(5)选取权重最大的K个词最为新的扩展查询词，与原始的查询词组成新的查询词集Q’；

(6)重复步骤3；

(7)输出查询结果，返回相关文本。

这种扩展查询方式有两个问题，其一是使用BM25方法进行查询词扩展，BM25方法是tf-itf模型的改进，它仅仅考虑词频的影响，而忽略了词义影响，对于同义不同词的情况，仅仅使用词频有时会对查询结果造成一些不好的影响，因此本文在BM25模型下，加入词义相似性判定，能获取更好的效果。其二，这种扩展方法没有将新的查询词的权重与原始查询词的权重进行区别，显然原始查询词才是用户最关心的，应该给予较高权重，而新的查询词不是用户最关心的词语，权重应该小一些，实验表明改进后的方法效果明显增强。

## 2.5 本章小结

本章首先对社交网络数据采集与查询系统的需求进行分析，分析出了系统主要功能点：服务器端需要数据采集功能、数据查询功能以及需要与其他功能进行交互，客户端需要有3个子界面，分别是数据采集子界面、数据查询子界面与状态监控子界面，而数据存储功能必须保证数据的可靠性、稳定性、安全性以及高并发性，同时需要快速的查询效率。

对于这个采集与查询系统而言，核心是服务器端中的数据采集方法与数据查询方法，对现有的数据采集技术加以分析提出了一种基于DOM树结构与URL链接模板的采集技术，同时对于基于自动相关反馈的查询扩展方法提出了改进，引入了词义相似性与查询词重定义权重，以此获取更好的查询结果。

# 第三章 社交网络自动化数据采集方法

## 3.1 背景

社交网络蕴含了海量数据，同时每年都有不同的社交媒体出现，如果按照传统数据采集方法，采集社交网络数据的工作者必须拥有大量的相关专业知识，如了解HTML格式、语法，熟悉一门数据采集程序，同时还必须了解数据库相关的内容等等，这大大加深了数据采集工作者的难度，并且传统的采集方式针对不同的社交网站需要逐一解析每个网站的网页HTML格式，这些重复性的工作将耗费工作者大量不必要的时间，更有甚者在网站每次改版后，以前的数据采集方法无法使用，需要重新构建。因此，如何简单、快捷、准确的采集不同网页的数据成为关键问题。

针对不同社交网站数据采集的问题，本系统的数据采集方法使用基于DOM树结构自动提取数据技术，即通过前后端交互，系统使用者不需要知道采集网页具体的HTML，也不需要了解HTML相关知识，仅仅将需要采集的数据文本的全部或者一部分传给后台的数据采集功能，此功能将会自动产生相应的数据采集路径。

针对扩展同构网页采集的问题，本文系统使用URL链接模板技术扩展采集网页的数量，只要使用者给定的初始采集网页拥有扩展链接，使用者不必知道链接的格式或者链接在DOM树所处的位置，数据采集方法根据网页匹配与采集内容匹配的算法，计算出最有可能的扩展链接，并生成链接扩展规则。根据生成的链接扩展规则，将当前采集网页中的所有链接与链接扩展规则进行匹配，即可快速扩展需要采集的链接数量。

## 3.2 方法思路

社交网络数据采集方法基于DOM树结构技术与URL链接模板技术，此方法确认采集数据时，属于半自动化，需要用户少量参与，而在扩展链接时，属于全自动化，能够快速有效的采集所有同构网页，因此本文的数据采集方法总体具有较强的自动化与自适应性，具体流程见图1-1。

首先，使用者在客户端的采集页面选择需要采集的网页，服务器端读取选定网页的HTML源码，并从源码中提取需要的信息，重构DOM树，然后，系统使用者将需要采集的数据内容发给后端的采集代码生成模块，经过内容的匹配与使用者的选择后，生成内容采集路径，接着在开始采集内容之前，通过链接扩展模块，生成链接扩展规则，最后，在采集内容的同时扩展链接，最终达到获取所有完整信息。



图1-1 社交网络数据采集流程图

## 3.3 重构DOM树

在重构网页DOM之前首先需要了解什么是DOM树，DOM是Document Object Model文档对象模型的缩写，根据W3C DOM规范，DOM是一种与浏览器，平台，语言无关的接口，使得使用者可以访问页面中其他的标准组件。DOM是以层次结构组织的节点或信息片断的集合，这个层次结构允许开发人员在树中导航寻找特定信息，由于它是基于信息层次的，因而 DOM 被认为是基于树或基于对象的。同时HTML DOM 定义了访问和操作HTML文档的标准方法，并把 HTML 文档呈现为带有元素、属性和文本的树结构(节点树)。

将网页HTML源码变为HTML DOM树的方法有很多，本文选择python提供的BeautifulSoup包。BeautifulSoup提供一些简单的、python式的函数用来处理导航、搜索、修改分析树等功能，它是一个工具箱，通过解析文档为用户提供需要抓取的数据，因为简单，所以不需要多少代码就可以写出一个完整的应用程序。同时BeautifulSoup自动将输入文档转换为Unicode编码，输出文档转换为utf-8编码，使用者不需要考虑复杂的编码方式。

重构网页DOM树具体步骤如图1-1所示。



图1-1 重构DOM树具体步骤

1. 使用BeautifulSoup将用户选定网页的HTML源码变为DOM树；
2. 将html标签视为初始节点；
3. 判断当前节点的子节点是否存在；
4. 子节点不存在，则将没有访问过的节点当成父节点，并重复步骤3；
5. 子节点存在时，获取节点标签、id、class、文本等信息；
6. 根据子节点与父节点的关系，构建自定义的DOM树；
7. 选择没有访问过的节点当作父结点，重复步骤3。

### 3.3.1 广度优先提取节点

广度优先算法(Breadth-First Search)，即广度优先搜索，又称作宽度优先搜索，或横向优先搜索，简称BFS，是一种图形搜索算法。换而言之，BFS是从根节点开始，沿着树的宽度遍历树的节点，如果发现目标，则演算终止。BFS是一种盲目搜寻法，目的是系统地展开并检查图中的所有节点，以找寻结果，它并不考虑结果的可能位置，而是彻底地搜索整张图，直到找到结果为止。

图1-1是某个网页的HTML源码，广度优先获取节点：首先将html标签看作初始结点，标记此节点为0编号，html标签有两个子节点head与body，分别标为1与2，然后先后找出1与2的直接子节点，并依次编号，比如body下只有一个子节点nav，那么将nav编号，并指出它的父节点为body，即2号编号。这样一层一层遍历，可将所有节点获取，并且可以获取每个节点的父节点，为其后构建DOM树提供信息支撑。



图1-1 某个网页的HTML源码

### 3.3.2 提取信息节点的标签及相关属性

为了构建新的DOM树，需要获取原始DOM树的必要信息，同时新增一些相关的有用信息，表1-1是重构DOM树的全部属性。

表1-1 新DOM树的全部属性

|  |  |
| --- | --- |
| 属性 | 说明 |
| FatherNum | 父节点编号 |
| SelfNum | 本节点编号，具有唯一性 |
| Tag | 标签 |
| Class | 标签的类值 |
| Id | 标签的ID值 |
| Content | 当前节点的文本 |

获取属性的伪代码如下所示：

循环获取父节点的所有子节点

{

如果子节点存在

{

获取子节点标签Tag

如果标签的类值或者id值存在

{

获取类值或者id值

}

否则

将其赋值None

获取该子节点的文本Content

子节点编唯一编号并获取父节点编号

}

否则

下一个子节点

}

### 3.3.3 生成DOM树

在上一节已经获取了需要的相关信息，将每个节点信息以json键值对的形式存入MongoDB数据库中，并使用自身编号建立索引，通过自身编号与父节点编号形成一颗新的DOM树，重构后的DOM树较原始DOM树在回溯路径时有着巨大的优势，即当确定某个节点的内容是用户所需内容时，新的DOM树能够快速的生成相对采集路径。重构后的DOM树如1-1所示。



图1-1 重构DOM树

## 3.4 自适应数据采集代码生成

自适应数据采集代码的生成需要用户参与其中，一共需要6步，如图1-1所示，首先用户需要在客户端的采集内容界面中输入或者粘贴需要采集的全部或者部分内容，通过请求传送给后端服务器，服务器端接收到需要采集内容后，需要对内容做预处理，使数据标准化，然后将预处理后的内容放入重构后的DOM树中进行正则匹配，接着将所有匹配成功的节点信息返回客户端(成功匹配的节点不一定只有一个，可能有多个)，客户端接收返回的信息并将其展示给用户，用户选择合适的节点，并将选择结果发送服务器端，服务器中的相应函数根据节点信息生成采集路径，并将其保存在数据库中。



图1-1 自适应数据采集代码生成流程

### 3.4.1 采集信息搜索

信息搜索即通过用户给定的采集内容定位DOM树中节点，服务器端在接收到采集内容后，需要预处理这些数据，预处理分为3步：首先将内容的回车改为空格，然后根据空格将整个内容分成小段内容，最后处理特殊字符，如“\\”，“\*”，“（”，“）”等，将这些特殊字符变为转义字符，为其后的正则匹配做好准备。

信息匹配有两种方式，一种是完全匹配(perfect match)，另一种是正则匹配(regular expression matching)，在信息匹配时，首先对内容进行完全匹配，完全匹配效率高，可以节约时间，但是有时用户给出的采集内容不全，比如有时用户需要采集某个新闻网站的正文，一般来说用户不会将全部需要采集的内容传给服务器端，仅仅传一部分，这时完全匹配无法成功，那么需要使用正则匹配，在之前的内容预处理时，已经将内容分成了许多小段，正则匹配每一小段，全部成功匹配，即是满足需求的节点。

### 3.4.2 采集信息定位

采集信息定位主要发生在客户端，如图1-1所示，用户通过选择按钮选择需要的节点。例如，用户输入采集内容A，经过服务器端匹配信息后，获取到3个满足需求的节点，这3个节点的文本都包含了内容A，这时服务器端需要分别返回每个满足需求节点的自身节点、父节点以及父节点的父节点，并将这3个节点的内容显示在内容框中，用户可以通过这些信息判断此节点是否为所需节点，如果不是，则点击下一页，继续进行判断，直到显示完3个节点信息，最后用户将确定的节点发送给服务器端，即完成了采集信息定位操作。

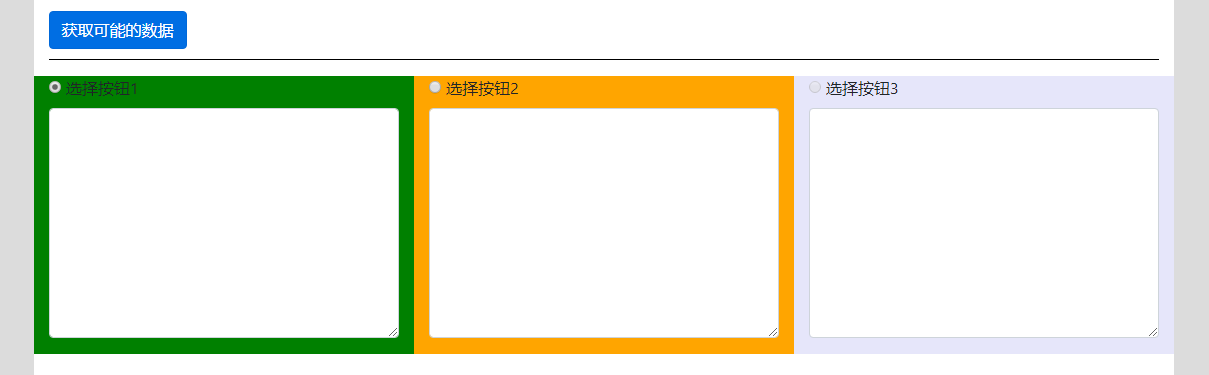


图1-1 用户采集界面的选择节点截图

### 3.4.3 信息路径构建

采集信息路径即为原始HTML DOM树从某个节点到采集信息节点的路径，这条路径包含每一步需要进行的操作以及相关属性，如图1-1所示。

表1-1 路径属性

|  |  |
| --- | --- |
| 属性 | 说明 |
| type | 操作类型，包括find，getText，getImg。 |
| tag | 当前节点下的子节点的标签名 |
| num | 有相同标签名时，说明标签名的位置 |
| class | 标签名的class值 |
| id | 标签名的ID值 |

为了增加自适应性，路径构建采用相对路径结合绝对路径的方式，传统的路径构建一般来说采用的是从html标签开始，一直到采集内容节点为止的绝对路径，这种方式对于当前的网页结构有时无法适用，现在的一些网页是动态生成的，样本网页的绝对路径不一定适用于某些动态网页，同时针对网页的小范围改动，例如在网页某个块中增加广告代码段，绝对路径也不能很好的适应，因此本文选用相对路径与绝对路径结合的方式。通过重构的DOM树，能够迅速的从绝对路径节点中选出具有唯一性的class或者id值，如果绝对路径中拥有这样的节点，那么采集路径可以直接这些节点开始，否则，路径选用绝对路径。

## 3.5 同构网页链接扩展

同构网页链接扩展的关键在于确认同构网页，即网页的相似判定。早在1998年，IF Cruz等人【IF Cruz ， S Borisov ， MA Marks ， TR Webb. Measuring Structural Similarity Among Web Documents: Preliminary Results. International Conference on International Confe... , 1998 , 1375 :513-524】就对网页的相似性进行了研究，他们主要针对网页的标签频率分布(Tag Frequency Distribution Analysis)，这种方法的相似性判断需要大量的样本，同时不同标签权重的获取需要对样本网页主题分类，以此来获取平均值。此方法对于特定网页结构相似性判定效果不理想，但这篇文章提供了一种标签判定网页相似的思路，因此，本文提出了一种新的基于标签与特定路径的相似性判定方法。

### 3.5.1 链接扩展规则

传统的判断网页相似性的方法基于标签频率分布，这种方式对于给定网页的相似性判定效果并不理想。对于同一网站，往往拥有几个板块，这些板块结构大同小异，而用户只需要采集其中一个板块的内容，这时，如果使用传统的方式，很难将不同板块的网页分开，阈值的设定变得异常困难。因此针对本文研究的同构网页相似性判定，使用了唯一class值标签与采集内容数量占比作为判断依据，认为同构网页的唯一class值标签基本一致，同时测试网页中能够采集内容的数量必须大于用户需要采集数量的五分之一。网页相似性定义公式如1-1所示：



其中， 表示当前网页获取的采集数量与用户需要采集的数量的比值， 表示以样本网页唯一class的标签数作为总数与当前网页唯一class的标签数所占比率， 与分别是与的权重，+ =1。

链接扩展流程如图1-1所示，首先需要在DOM树中获取所有唯一class值的标签，例如对于网页HTML块“<span class="year">(2018)</span>”，其中整个网页的span标签只有这一个的class值为year，那么span-year将看作一个键值对，然后获取样本网页的全部URL链接，并将链接存入一个队列中，接着依次访问每个URL链接，获取它的HTML源码并转换为DOM树，这时需要分别统计所有唯一class值的标签以及统计能够获取的采集内容的数量，根据公式1-1计算网页的相似性，将全部URL链接网页的相似性进行排序，选出相关性最大且长度相同的两个URL链接，最后将这两个链接以“/”分成几段，对比每一段的值，找出共性与异性，并生成扩展链接规则。



图1-1 扩展链接流程图

### 3.5.2 测试与分析

图1-1是 （准备采集3-4个网站：CSDN，豆瓣电影、微博、领英）

图片包含 屏幕截图

已生成极高可信度的说明

图1-1 起点网站

测试链接规则满足可扩展性、可靠性、唯一性、随机性。

## 3.6 本章小结

本章详细描述了社交网络数据采集方法的实现流程，此方法主要基于DOM树结构技术与URL链接模板技术，相对于传统的DOM树技术，本文自定义了DOM树，此DOM树简化了原DOM树的同时增加了节点编号以及父节点信息，使新的DOM树能够快速检索信息与生成采集路径。对于采集路径的生成方面，本文提出了绝对路径结合相对路径的方法，此方法增加了采集路径的自适应性，对一些动态生成的网页有着较好效果。最后同构网页链接扩展规则的提出，使得整个社交网络数据采集方法变得普遍性、通用性以及完整性，用户几乎不需要任何专业知识，仅仅通过选择需要采集内容的文本就可以完整的采集所有同构网页的数据。

# 第四章 基于权重与语义的扩展查询方法

## 4.1 背景

（系统引申出查询——查询扩展意义——缺点——提出新的解决方法）

如今的21世纪，信息无疑是既廉价又宝贵的资源，廉价在于每天互联网上都会产生无法想象的巨大量的数据，而宝贵在于在这些海量数据中隐藏有可以转化为千金的信息。大数据一词正是基于现代社会互联网巨大信息，它的定义如下：一种规模巨大的数据集合，具有海量的数据规模、快速的数据流转、多样的数据类型和价值密度低四大特征。对于社交网络数据而言，正是具备大数据的特征——大量、高速、多样、低价值密度、真实性。仅仅获取海量社交网络数据，而不加以处理、分析，这些数据的价值将会大大降低，因此用好存储在数据库中的数据在于提高对数据的“加工”能力，通过“加工”实现数据的增值。在本文中对于已采集数据的有效、快速查询就是对数据的“加工”。

本文的测试集来源于twitter，对于某次查询“英国广播公司世界服务部裁员”事件，用户查询词组为“BBC World Service staff cuts”，表1-1中全部是满足用户需求的文本，其中查询词用红色字体。

表4-1 部分满足用户需求的文本

|  |
| --- |
| world service cuts why we need the bbc |
| closure of bbc radio service information era tragedy eurasia review outlookclosure of bbc radio service inf |
| bbc exodus huge bill for moving shows and staff around uk pic |
| bbc inhouse faces 15% cuts as pattern emerges money |
| hsbc recognises and rewards long service staff the brunei times |
| bbc cut backs are a pain |

表1-1中一共有6句话，前两句中包含了大部分的查询词，而后4句仅仅包含一小部分查询词，特别是最后一句话只含有“bbc”一个查询词，如果使用传统的词频查找文本，那么后4个文本将较难被筛选出来，同时例如最后一句话“bbc cut backs are a pain”中的cut本应该是查询词，但是由于词性原因，将不被认为属于查询词，因此首先在查询文本之前，需要进行文本与查询词组的预处理，包括词性还原、统一大小写等，由于基于自动相关反馈的查询扩展方法使用的是BM25模型进行查询扩展，不能很好的解决查询词过少的情况，因此本文提出了基于自动相关反馈的查询扩展的改进方法——WRFSS(weighted revelant feedback and sentence similarity)算法，在原有的BM25查询模型基础上加入了语义相似性，同时在扩展查询文本时，引入了查询词权重，获得了更好的查询效果。

## 4.2 基于权重与语义相似性的相关反馈方法

### 4.2.1 方法思路

WRFSS算法的基本思想是：利用查询词与文本的词频与语义相似性，获取前N个文本（本文取前30个），同时假定这些文本有效，并根据文本逻辑相关筛选出较重要的词作为扩展查询词，将这些查询词与原始查询词一起组成新的查询词组，并使用基于查询词权重的词频与语义相似性重新查找文本，提高文本全面性。

基于WRFSS算法的扩展查询流程如图1-1所示。



图4-1 基于WRFSS算法的扩展查询流程图

基于WRFSS算法的扩展查询流程具体步骤如下：

(1)用户输入初始查询词集合Q以及查询文本集合D；

(2)将Q与D转化为相关向量与，其中，；

(3)使用BM25模型与语义相似性检索文本集，计算每个文档与初始查询词组的相似性，并将这两种相似性根据公式1-1计算出一个新的相似性同时按大小降序排序；

(4)取出前N个文本，将这些文本的所有词作为扩展的候选词；

(5)利用词频特性的公式1-1计算出候选词的权重，并降序排序；

(6)文本按照1：1的方式，选择与查询词组相同个数的前M个候选词；并与原始查询词组成新的查询词组Q’；

(7)利用公式1-1，计算出新的查询词组每个查询词权重；

(8)使用基于查询词权重的BM25检索模型与语义相似性，按照步骤3重新对文本集进行相似性计算并降序排序；

(9)最后取出结果集的前N项文本，输出结果。

### 4.2.2 方法描述

WRFSS算法的核心是关于扩展查询中的新查询词与原始查询词需要赋予不同的权重以及在传统的词频查询的基础上结合了语义相似性，基于权重的BM25算法与语义相似性方法具体如下：

(1)基于权重的BM25算法

在提出基于权重的BM25检索模型前，首先需要了解TFIDF与 BM25算法。

TFIDF是由Salton等人【Salton, G.; Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. Information Processing & Management. 1998,24 (5): 513–523.】在1988 年提出的，其中TF(term frequency) 称为词频, 用于计算该词描述文档内容的能力；IDF(inverse document frequency)称为逆向文档频率, 用于计算该词区分文档的能力。TFIDF的主要思想是：如果某个词或短语在一篇文章中出现的频率TF高，并且在其他文章中很少出现，则认为此词或者短语具有很好的类别区分能力，适合用来分类。具体公式如下：





其中为索引词的在文本中的数量，n表示总的文本数量，表示在文本中的词频，表示的逆向文本频率。

但是TFIDF算法有一个严重的问题，它仅仅认为一个词的词频越小就越重要，反之就越无用，显然这并不是完全正确的，它没有考虑到文档的长度，例如，某个查询词Q在一个长度为10000的文本(拥有10000个词语的文档)出现了10次，同时这个查询词在一个长度为50的文本中出现了5次，由于Q在后一个文本中比例比前一个高，因此后一个文本理所应当排在前面，但是Q在前一个文本中出现的次数更多，按照TFIDF公式计算，前一个文本的权重反而高于后一个文本，者就会造成严重的误差。

为了解决上诉问题，Robertson等人【SE Robertson,S Walker.Some simple effective approximations to the 2–Poisson model for probabilistic weighted retrieval.Springer London , 1994 :232 - 241】提出了TFIDF的改进算法，即BM25算法，BM25算法一般公式如下：



其中Q表示查询词组，表示其中一个查询词，d表示代查询的文本，表示的权重(与词频相关)，表示查询词与文本d的相关值。

一般的计算使用的是IDF，公式如下：



其中，N表示待检索的文本集数量，表示包含了查询词的文本数量。  
 BM25算法中相关性值一般形式如下：





其中，，，b为调节因子， 是查询词在文本d中出现的频率，是文本d的长度，是所有文本的平均长度，是查询词在查询词组出现的次数，通常由于查询词只会出现一次，因此的值设置为1，这时的公式变为：



BM25算法解决了文本长度的问题，但是当BM25用在自动相关反馈的查询扩展上时，由于新加入的查询词并不是用户直接指定的查询词，理论上，这些新的查询词在扩展查询的时候权重应少于原始查询词，但是BM25算法对于词的权重仅仅依赖于IDF，如果新的查询词的稀有度大于了原始查询词，那么查询结果将会受到影响，为解决此问题，引入了基于局部上下文分析的方法。

基于局部上下文分析方法将查询词组Q认为一个整体，并在局部本文集中计算相似性，获取最相关的词语进行查询词扩展【】。计算相似性具体公式如下：





其中表示局部文本集中的某个词语，表示第i个查询词，表示查询词在文本j中出现的次数，表示词在文本j中出现的次数，而与分别表示基于部分文本集的查询词与词的IDF值，是一个调节因子，通常取值0.1。

通过局部上下文分析方法可以计算出新查询词对于原始查询词的权重，同时将拥有权重的查询词引入BM25模型，这样扩展查询可获取更为准确的文本结果。

(2)语义相似性方法描述

使用语义相似性判别句子相似性的算法有很多，经典的方法有word2vector与wordnet，但是这些句子相似性方法主要以下挑战：

1. 两个语义相同的句子之间有词汇的差异(注意这里不是指近义词)。例如表1-1中的E1与E2同义但不同词典，红色部分表示同一个意思，但E2是由短语表达了“不相关”这个含义，这与以往的近义词不同。
2. 语义相似性应该在不同粒度(词语、短语与语法)中进行测量。例如，观察表1-1，E2中的“not related”在与E1中的“irrelevant”进行匹配时是不可分割的短语。
3. 两个句子之间的差异信息也很重要。例如，观察表1-1中E3，E4，E5这三个句子，通过判断不相似的部分，能够很容易的判断出E3与E5都表达了同一个意思——“这是一项关于鲑鱼的研究”(其中“scokeye”属于鲑鱼家族，而“flounder”不是)，但是E4不同于E3，主要强调的是“研究关于红色的鲑鱼(一种特殊的鲑鱼)”而不是仅仅研究鲑鱼，因为“sockeye”与“coho”都是鲑鱼的一种，强调研究前者，否定研究后者，这就与E3的意思不一样了。

表4-2 用于表述句子相似性挑战的例子

|  |  |
| --- | --- |
| E1 | The research is irrelevant to sockeye. |
| E2 | The study is not related to salmon. |
| E3 | The research is relevant to salmon. |
| E4 | The study is relevant to sockeye, instead of coho. |
| E5 | The study is relevant to sockeye, rather than flounder. |

为了解决上述问题，Z Wang【Z Wang ， H Mi ， A Ittycheriah.Sentence Similarity Learning by Lexical Decomposition and Composition.IBM T.J. Watson Research Center Yorktown Heights, NY, USA,1997.】等人在2017年提出了通过词汇分解与组合的方式来判断句子的相似性。为了解决第一个挑战，需要将每个单词表示为分布式矢量，这样就可以计算形式上不同但是语义相关的词的相似性，对于第2个问题，可以假设每个单词在语义上与其它句子中的几个单词相匹配，并根据所有单词向量计算每个单词的语义匹配向量，最后，为了解决挑战3，假设每个单词可以部分匹配，并且基于语义匹配向量可以分解成相似与不相似两个部分，在用这两部分去匹配研究相似性。具体流程见图4-2。



图4-2 基于词分解与组合的语义相似性

1)句子向量化

通过提前训练获得大量单词的向量，本文使用了一个已经训练完成的100000\*200(100000个单词，每个单词维度为200)的向量组，通过向量组中查找获取，将句子S与T变为句子向量，，  ，n与m分别是句子S与T的长度，每个单词的维度为d，d=200。

2)语义匹配

使用S与T，利用公式1-1计算出单词相似矩阵 。



公式1-1中， 表示句子S中的单词i向量， 表示句子T中的单词j。

在获取了单词相似矩阵后，接着将S中的每个单词作为一个单元，通过在另一个句子T中合成部分或全部单词向量，用于计算出每个单词 的语义匹配向量 ，同时与之相对应的有 的语义匹配向量。公式见1-1：



公式1-1中，，即首先获取单词相似矩阵第i行中最大数的j，然后等于T中的单词。

3)分解句子

分解句子基于语义匹配向量，将其分成两个部分：相关与不相关。首先，需要检测是否有一个词在另一个句子中完全匹配或者是否与相等，如果满足，则向量属于相关部分并且相对应的不相关部分赋值0向量，反之，将分配给不相关部分，相关部分赋值0向量，具体公式见1-1：



4)组合句子

本文使用一个双通道神经卷积网络(CNN)和一个过滤器实现句子的组合。CNN模型最重要的两个操作一是卷积操作，二是最大池，卷积操作定义了一个d\*h的过滤器，其中d为单词的维度，h是过滤器窗口长度，每次过滤都需要将相关与不相关部分结合起来，具体公式见1-1：

 (1-1)

显然，只需要将 与 对应位置的数相乘就可以得到组合后的特征向量。然后使用最大池，将两个特征的维度变为同一维度。

5)相似评估

求两个维度相同的向量的相似度，最常见的是余弦相似度方法，具体公式见1-1所示：

 (1-1)

通过以上五个步骤——句子向量化、语义匹配、分解句子、组合句子以及最后的相似评估，最终实现两个句子的语义相似性判定，获取相应的评分。

## 4.3 扩展查询方法实现

### 4.3.1 预处理

扩展查询方法的第一步即是查询词与查询文本的预处理。预处理分为两个部分，其一是对查询词与查询文本大小写统一，大小写不统一词频数值会造成一定影响，甚至无法获取到正确的结果，例如，用户输入查询词“bbc”，希望获取有关BBC的信息，但是查询文本集中都是以大写存在，即“BBC”，这种情况下，就很难查询到用户想要的信息。其二是词形还原，词形还原包括动词过去时、过去分词还原为动词原形，名词复数还原为单数，形容词副词形式还原为形容词原型等等，词形还原的目的在于扩大查询范围，不漏掉查询词的任何形式。例如，查询词是“kill”时，使用了词形还原后，范围就扩大到了killed、killing、kills等。本文对于词形还原使用了python提供的nltk包中的WordNetLemmatizer类，这个类方便、快捷、有效，实验结果得到显著提升。

### 4.3.2 文本筛选

文本筛选主要基于BM25模型与词分解与组合的语义相似性相结合的方法，查询词词组分别用BM25模型与语义相似性计算与文本的相关分数，将这两种分数归一化，并使用公式1-1计算文本最终的相似性。



其中表示文本i在BM25检索模型下获得的分数，表示文本i在语义相似性的检索中获取的分数，与查询词长度有关，具体公式如下：



其中是调节因子，经过大量测试，本文取0.1，d表示查询词长度，同时。

本文主要针对社交网络短文本，一般用户输入查询词词组是希望获取某个事件相关的文本或者希望获取有意义的文本，但是文本筛选时可能会出现只有1-2个单词的句子，这种句子往往没有具体的含义，无法清晰的表达语义，因此本文将长度小于3的文本排除，同时文本筛选的数量定位30个。

### 4.3.3 查询词扩展

WRFSS算法的查询词扩展基于自动相关反馈的查询扩展方法中的查询词扩展，认为扩展词与部分文本集中出现的频率有关。

扩展词的权重计算公式如下：



其中N表示上一步中筛选的前N个文本，表示文本集中剩余的文本个数，是调节因子，当的值为0时表示自动反馈，的公式如1-1所示：



其中表示文本i的长度，表示整个文本集的平均长度。

通过对进行降序排序，本文按照1:1选出除原始查询词的前n个词作为新的查询词，并与原始查询词组成新的查询词词组。

### 4.3.4 再次检索文本

再次检索文本需要计算出查询词的权重并使用增加了查询词权重的BM25模型且再次结合语义相似性，基于查询词权重的BM25具体公式如：



其中表示其中一个查询词，表示查询词的逆文本频率值，表示查询词在整个查询词词组中的权重，本文定义原始查询词的权重为1，中新加入的查询词的公式如1-1所示：



公式 1-1 中的 表示新加入的查询词的权重，表示原始查询词与的相关性，表示原始查询词中的每个查询词的权重。

这次查询检索的结果作为最终结果，如同文本筛选一样，降序排序，选出前N个文本返回给用户。

## 4.4 测试与分析

### 4.4.1 评价标准

在测试之前需要确定评价标准，信息检索评价主要根据是否符合用户需求进行判断，因此在测试前用户需要对测试集进行人工标注，本文以经过标注的推文作为文本测试集，满足用户需求的推文标注为1，不满足的推文标注为0。

信息检索主要评价方式有5种：1)召回率(Recall)；2)准确率(Precision)；3)F1；4)平均准确率（mean Average Precision）；5)前K项准确率。



图4-3 信息检索说明图

召回率：指的是检索到的相关文档与所有相关文档的比值，即A/(A + C)。

准确率：指的是检索到的相关文档与检索到的所有文档的比值，即A/(A + B)。

F1：准确率与召回率的加权调和平均数，公式见1-1。



平均准确率：用于解决召回率、准确率与F-measure的单点值局限性，公式如下：



其中P(r)指的是r点的准确率。

前K项准确率：当信息检索的结果被排序时，可以只考虑前k个结果并计算这些结果的精度，这就是在K点的准确率，即P @ k。

### 4.4.2 测试与分析

本文使用的测试集是推文短文本。使用TwitterAPI采集推文数据，数据的格式是json，获取的推文使用数据存储功能中的MongoDB存储模块进行存储，图1-1显示的是其中一条推文在软件Robomongo中存放的形式。本文测试一共采集了49个话题共50000条推文，并对每条推文根据与自身话题的相关性进行了人工标记，如果与话题相关则标记为1，反之标记为0。

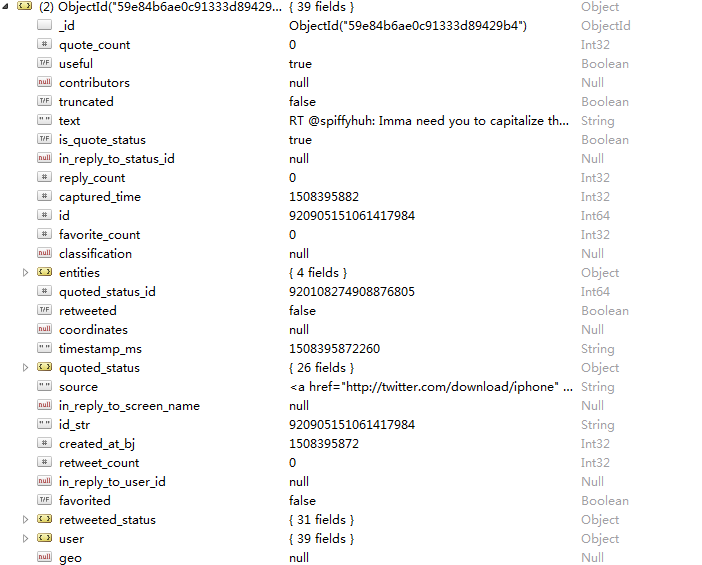


图4-4 一条原始推文

推文部分重要字段的含义见表1-1。

表4-3 推文部分字段的含义

|  |  |
| --- | --- |
| 字段 | 含义 |
| Id | 推文唯一编号，与时间有关，时间越大id越大 |
| Text | 推文的短文本 |
| Created\_at | 推文创建时间 |
| User | 发布这条推文的用户信息 |

本文以2011年关于美国国家标准与技术研究院(NIST)引导的计算机安全话题为例进行具体测试分析。用户输入关键词词组“NIST computer security”,第一次查询分别使用BM25方法与WRFSS方法，获得前6条查询文本结果分别如表1-1与1-2所示，表中第一列表示这条推文是否相关(1表示相关，0表示不相关)。从表1-1与1-2的结果可以发现，表中前4个文本完全一样，说明文本查询主要还是以查询词的词频为主。接着，由于WRFSS方法引入了语义相似性，削弱了词频的权重，说明语义相似性一定程度上增加了语义对于查询的重要性，正是如此，才有了 “security news nist issue cloud security guideline government standard body ha launched wiki”这句话在表1-2中得分高于表1-1。然后，WRFSS方法使得某些含查询词较少的文本获得了较高得分，表1-2中倒数第二个文本中仅仅包含一个查询词“nist”，这种文本一般在使用BM25方法进行查询时，很难被获取，而加入了语义相似性判定的WRFSS方法能够将语义与查询词词组相关的文本提取出来，这就能够获得更好的查询效果。

表4-4 使用BM25第一次查询获取的前6条查询文本

|  |  |
| --- | --- |
| 0 | nist address security threat challenge information security program b2b |
| 1 | security news nist issue cloud security guideline government standard body ha launched wiki |
| 1 | nist guidance tackle public cloud security |
| 1 | nist guidance tackle public cloud security bankinfosecurity |
| 1 | nist appoints nstic senior advisor secureidnews press release |
| 0 | nist doe leading definition i thinking bottom stack data center computer model |

表4-5 使用WRFSS第一次查询获取的前6条查询文本

|  |  |
| --- | --- |
| 1 | security news nist issue cloud security guideline government standard body ha launched wiki |
| 0 | nist address security threat challenge information security program b2b |
| 1 | nist guidance tackle public cloud security |
| 1 | nist guidance tackle public cloud security bankinfosecurity |
| 1 | nist guidance tackle public cloud security 2 special pub cloud defs virtualization |
| 1 | nist report aim u agency deploy cloud apps ease cloud computing service deplo |

表4-6与4-7分别表示使用BM25方法与WRFSS方法第二次扩展查询获取的前6条查询结果。从总的相关文本来看，WRFSS方法明显优于传统的BM25方法，在使用两种查询方法之前，它们的扩展查询词都是“cloud tackle public”这三个词，表4-6中有两个不相关文本——“cloud guideline security privacy public cloud computing business edge blog cloud”与“cloud service hp enters public cloud market hybrid solution hp enterprise cloud servicescompu tcn”，这两个文本都有一个特点：包含了太多的扩展查询词而原始查询词较少甚至没有，造成这样的原因是传统扩展查询中BM25方法没有考虑新的查询词权重的问题，而用户期望的查询结果必定与原始查询词有较大关联，WRFSS方法引入查询词权重解决了此问题。

表4-6 使用bm25第二次查询获取的前6条查询文本

|  |  |
| --- | --- |
| 1 | nist guidance tackle public cloud security 2 special pub cloud defs virtualization |
| 1 | nist guidance tackle public cloud security |
| 1 | nist guidance tackle public cloud security bankinfosecurity |
| 0 | cloud guideline security privacy public cloud computing business edge blog cloud |
| 0 | cloud service hp enters public cloud market hybrid solution hp enterprise cloud servicescompu tcn |
| 1 | nist report aim u agency deploy cloud apps ease cloud computing service deplo |

表4-7 使用WRFSS第二次查询获取的前6条查询文本

|  |  |
| --- | --- |
| 1 | nist guidance tackle public cloud security 2 special pub cloud defs virtualization |
| 1 | nist guidance tackle public cloud security |
| 1 | nist guidance tackle public cloud security bankinfosecurity |
| 1 | security news nist issue cloud security guideline government standard body ha launched wiki |
| 1 | nist report aim u agency deploy cloud apps ease cloud computing service deplo |
| 1 | nist issue cloud security guideline government standard body ha launched wiki feedback |

对于信息查询结果，由于用户时间与经历有限，往往最关心前N条信息，因此本文采用前K项准确率(P@k)评判查询方法的好坏。图4-5展示了传统的基于自动相关反馈的查询扩展方法与WRFSS扩展查询方法对测试集的最终测试结果，从图中数据可知，传统的扩展查询方法P@10的值为0.492，WRFSS扩展查询方法P@10的值为0.541，准确率提高了10个百分点；而P@30本文的方法比传统的方法准确率提高了18个百分点，这两个结果充分表明了WRFSS扩展查询方法比传统的自动相关反馈的查询扩展方法效果高出很多。

图片包含 屏幕截图

已生成极高可信度的说明

图4-5 传统扩展查询与WRFSS扩展查询的对比图

## 4.5 本章小结

本章首先研究发现基于自动相关反馈的查询扩展方法虽然自动化较强，但是存在两个问题，一是很难查询到含较少查询词的文本，二是没有考虑新旧查询词在第二次查询中权重的问题。因此本文针对这两个问题，提出了基于权重与语义的扩展查询方法，并分别对增加权重与语义相似性方法进行了详细描述，对于查询词权重使用基于局部上下文分析方法，通过局部文本计算出新查询词的权重，对于语义相似性，本文通过词汇分解与组合的方式来判断句子与查询词词组的语义相似性，并将权重与语义融入基于自动相关反馈的查询扩展方法中，得到WRFSS扩展查询方法。最后通过测试分析，得出了新的方法优于传统的扩展查询方法的结论。

# 第五章 系统测试与分析

## 5.1 测试环境

测试环境包括数据存储环境与数据采集查询(包括客户端)环境。以下是对这两种环境的说明。

数据采集查询环境：1台安装有windows764位操作系统的台式机，内存8G，CPU为Intel Core i5-6600，主频3.30GHz。这台测试机上安装的软件包括：1)python2.7；2)google浏览器；3)python依赖库包括：nltk、Django、pymongo、elasticsearch、gensim、selenium、bs4、requests等。

数据存储环境：一共3个存储节点，主节点CPU为Intel Xeon E7-4809 v3，主频为2.00GHz，含有2个物理CPU，共16个逻辑核，内存为330GB，数据存储的磁盘为16T的共享磁盘阵列，操作系统是Red Hat Enterprise Linux Server release 6.8，内核为64位的Linux 2.6.32。安装的软件包括：1)拥有3个分片的MongoDB3.4.3；2) ElasticSearch-5.6.3。

## 5.2 社交网络数据采集功能测试与分析

可以在前面加一些图片

### 5.2.1 测试

### 5.2.2 分析

主要是分析系统的能力（与前面不同），如采集速度能力，持续采集能力、分析为什么采集的慢，怎么提高等等

## 5.3 社交网络数据查询功能测试与分析

### 5.3.1 测试

### 5.3.2 分析

## 本章小结

# 第六章 总结与展望

## 6.1 工作总结

社交网络数据采集与查询系统在海量数据获取上有着巨大的优势。如今互联网中的数据都隐藏在各式各样的网页结构中，人们想要获取数据既要针对性的研究网页格式，又要学习相关的网络专业知识，既耗时又耗力，而本文的采集系统解决了这个问题，用户只需要一台电脑并懂得一些常识性的知识就可以全面、完整的采集互联网网页中任意的数据与信息。

同时系统在巨量数据的查询上也有着一定的优势。一般而言，用户需要从采集的数据中通过输入一组关键词词组以获取某个事件的相关数据，这时，信息查询往往面临两个问题——查询速率与查询准确率。在速率方面，本文通过构建高性能查询数据库，通过分片与索引的合理建立，充分利用服务器的高性能，实现了快速查询的目的。在准确率方面，本文以自动相关反馈的查询扩展方法为基础，引入了语义相似性，同时在常用的BM25方法引入了查询词权重，使得扩展查询中BM25方法前后两次的公式有细微的差别，这更好的应对了扩展查询的实际情况，整体上提高了信息查询的准确率。

本文具体的工作如下所示：

(1)研究了社交网络数据采集方法技术——基于统计理论的技术、基于视觉特征的技术、基于DOM树结构的技术、基于模板的技术。同时还研究了社交网络数据查询方法的意义以及数据查询的发展历史。

(2)设计了社交网络数据采集与查询系统。为了解决界面操作与数据展示、前后端交互、数据采集、数据查询和数据存储这5个功能需求，系统包含了3个功能模块——服务器端功能模块、客户端功能模块以及数据存储功能模块，其中系统的核心是服务器端的功能，而服务器端功能模块的核心是数据采集与数据查询方法，通过对数据采集与查询方法的分析分别提出了社交网络自动化数据采集方法与基于权重与语义的扩展查询方法。

(3)社交网络自动化数据采集方法的实现经历了3大步骤，一是完成了重构DOM树，为接下来分析、提取、扩展做铺垫；二是完成了自适应数据采集代码的生成，为了增加采集网页的自适应，本文提出并实现了基于相对路径结合绝对路径的采集路径生成方法；三是完成了同构网页链接扩展，通过比较网页相似性，获取扩展链接，接着生成链接扩展规则，实现快速扩展同构URL链接数量，同时经过扩展链接方法的测试，说明了扩展链接方法具有可扩展性、可靠性、唯一性以及随机性。

(4)实现了基于权重与语义的扩展查询方法，该方法在自动相关反馈的查询扩展方法上进行改进，在原本基于词频扩展的基础上引入了基于词汇分解与组合判断语义相似性的方法，同时基于局部上下文分析方法计算出了扩展查询词与原始查询词的不同权重，将这个权重融入原有查询模型，大大提高了查询结果的准确率。

(5)对社交网络数据采集与查询系统的数据采集功能、数据查询功能以及系统信息监控功能进行了一系列测试，验证了上述方法的有效性，同时说明了系统的实用性。

## 6.2 工作展望

本文设计实现了社交网络数据采集与查询系统，同时改进了传统的数据采集与查询方法。然而，由于自身的能力水平有所不足以及时间上的限制，对于系统的数据采集与查询方法依然可以继续改进、优化。这些能够改进的地方具体包括：

(1)数据采集方法每次采集主要针对同构网页进行采集，在采集数据的期间，网页可能发生格式上的变化，如果仅仅只有一小段变化，例如，网页增加了广告或者独立于以前的内容增加一个块内容，本文的采集方式由于使用了相对路径结合绝对路径的方式，因此可以应对此类问题。但是，如果网页的变化太大，例如，在2017年4月，领英的用户信息网页全部重新设计，整个HTML源码完全变化，面对这种情况，当前的数据采集方法无法处理，不过通过研究内容的相似性，发现虽然网页格式有着巨大的变化，但是采集的内容基本不变，因此利用这一特性，可以通过以前采集的内容，重新定位采集内容并构建采集路径。

(2)对于数据的查询方法，本文提出了基于查询词权重与语义相似性的扩展查询方法，在查询词扩展时，本文直接使用了自动相关反馈的查询扩展方法中的查询词筛选方法，这仅仅考虑了词频而没有考虑词的语义相似性，因此，今后可以在查询词扩展时引入权重，即对于待扩展词首先根据与原始查询词词组语义或者词义相似性而赋予不同的权重，接着使用词频的方式进行筛选扩展词，最后得到加权后的待查询词并选出前N个词。

# 致 谢

光阴似箭，日月如梭，三年研究生求学生涯即将结束，心中万分感慨与满满的回忆。在这三年里，我有幸结识了一批在研究生生涯中对我很重要的人，他们在我的学业、科研和研究生的生活方面给了我很多指导，帮助我不断成长，让我在学业上取得成功，在生活中变得成熟，也领悟到了一些人生哲理。

首先，我要对胡光岷老师与钱峰老师表达最为真挚的感谢。我正是被这两位老师渊博的学识以及对新知识得不断探索的精神所打动，坚定了我投身科研、奉献自我的决心。同时，这两位老师在科研中也给了我许多鼓励，我永远也忘不了在我研二攻克一项科研难题但始终不得入门时，胡光岷老师将我急躁的心情看在心里，但他并没有告诉我怎么解决这个问题，而是让我放平心态，告诉我心急是无法解决问题的。我用心体会这其中的哲理，渐渐不再一味的快速追求结果，而是以积极乐观平和的心态去研究发现科研中遇到的每一个小的难点，最后很轻松的解决了以往我认为我不可能解决的问题。

其次，我要衷心感谢费高雷老师。费老师作为我的指导老师，让我从一个研一对科研完全不懂的小菜鸟到研三随意给我一个科研任务我都有方向、有思路去一步一步解决问题。同时，他对我毕业论文的选题、创新点的确立、实验仿真、论文的撰写等方面都给予了悉心指导。忘不了费老师最开始将我提出的创新点批评的体无完肤到对我的创意露出满意的笑容。

接着，我还要真诚感谢教研室里的其他老师和每一位同学。成老师，我会时常想起我们一起共同研究采集系统的时光；鲁老师，您开怀的笑容真的给我很深的印象；姚老师，我时刻都会用您严禁的科研态度对待工作与生活；于老师，真的，您丰富的人生经历就如同一卷书，总是不禁让我细细品读；赵海林、谢星辰、郑夏、周成阳…对我来说你们都是我的兄弟姐妹，永远也忘不了我们相处的每一寸时光。未来之路漫漫，衷心祝愿教研室每一位老师身体健康，工作顺利；祝每一位同学前程似锦、事业有成。

然后，我要感谢我的父母在我16年的求学生涯中给与我最大的支持，是您们在背后默默的支持，才有了我这只幼鸟展翅高飞。

最后，感谢所有评阅老师在百忙之中给予我的指导和评阅意见，正是有了您们的意见才有了我这篇论文的最终成型。

杨杰

2018年3月于电子科技大学

# 参考文献

1. 中国互联网络发展状况统计报告.北京:中国互联网络信息中心, 2017年8月
2. http://data.weibo.com/report/reportDetail?id=404
3. 汪全莉,陈代春.Web数据挖掘在网络教育中的应用.《中国科技资源导刊》, 2008 ,40 (2) :44-47
4. 卢明泰.WEB数据挖掘及其在社交网络的应用研究.电子科技大学,2012
5. 黄琴.Web数据挖掘的研究及其在网络学习个性化推荐中的应用.湖南大学,2004
6. 云计算白皮书[R].北京:中国信息通信研究院, 2016年9月
7. S. Ghemawat, H. Gobioff, S. T. Leung. The Google File System[C]. Proceedings of the 19th ACM symposium on Operating systems principles (SOSP’03), ACM, 2003, 37(5): 29-43
8. J. Dean, S. Ghemawat. MapReduce: Simplified Data Processing on Large Clusters[C]. Proceedings of the 5th symposium on Operating systems design and implementation(OSDI’04). USENIX, 2004, 51(1):137-150
9. F. Chang, J. Dean, S. Ghemawat, et al. Bigtable: A Distributed Storage System for Structured Data[C]. Proceedings of the 7th symposium on Operating systems design and implementation (OSDI’06). USENIX, 2006, 205-218
10. Canalys. Media alert: Cloud infrastructure market up 49%, intensifying global data center competition[EB/OL] https://www.canalys.com/newsroom/media-alert-cloud-infrastructure- market-49-intensifying-global-data-center-competition, Canalys, 6th February 2017
11. D. Harris, S. Harris. Digital design and computer architecture[M]. MA, USA: Elsevier, 2012. 103-105
12. R. Stevens, S. Rago. Unix环境高级编程(第3版)[M]. 戚正伟, 张亚英, 尤晋元译. 北京:人民邮电出版社, 2014年6月, 377-380
13. J. Rosenberg. How Debuggers Work: Algorithms, Data Structures, and Architecture[M]. Canada: John Wiley & Sons, 1996, 21-34
14. 尤海鹏.基于Ganglia的数据中心监控平台设计[D].山东:山东大学, 2014
15. 王飞.基于分布式Nagios体系的应用监控平台设计与实现[D].上海:复旦大学, 2011
16. 陶芬,尹芷仪,傅建明.基于系统调用的软件行为模型[J].计算机科学, 2010, 37(4):151-157
17. A. S. Tanenbaum, D. J. Wetherall. 计算机网络(第5版)[M]. 严伟, 潘爱民译. 北京:清华大学出版社, 2012年3月, 15-17
18. Microsoft Technet.Virtual Private Networking: An Overview[EB/OL]. https://technet.microsoft. com/en-us/library/bb742566.aspx, Microsoft, September 04, 2001
19. D. Patterson, G. Gibson, R. Katz. A case for redundant arrays of inexpensive disks(RAID)[C]. Proceedings of the 1988 ACM SIGMOD international conference on Management of data (SIGMOD'88), ACM, 1988, 17(3):109-116
20. D. Borthakur. HDFS Architecture Guide[EB/OL]. http://archive.cloudera.com/cdh/3/hadoop- 0.20.2-cdh3u6/hdfs\_design.pdf, Hadoop Apache Project, 2008
21. F. Nielson, H. R. Nielson, C. Hankin. Principles of program analysis[M]. Berlin:Springer, 2015
22. N. Nethercote, J. Seward. Valgrind: A Framework for Heavyweight Dynamic Binary Instrumentation[C]. Proceedings of the 28th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI’07), ACM, 2007, 42(6): 89-100
23. C. Keung, L. Robert, C. R. Muth, et al. Pin: Building Customized Program Analysis Tools with Dynamic Instrumentation[C]. Proceedings of the 2005 ACM SIGPLAN conference on Programming language design and implementation (PLDI’05), ACM, 2005, 40(6): 190-200
24. S. Forrest, S. A. Hofmeyr, A. Somayaji, et al. A Sense of Self for Unix Processes[C]. Proceedings of the 1996 IEEE Symposium on Security and Privacy, IEEE, 1996: 120-128
25. S. Hofmeyr, S. Forrest, A. Somayaji, et al. Intrusion Detection using Sequences of System Calls[J]. Journal of computer security, 1998, 6(3): 151-180
26. A. Wespi, M. Dacier, H. Debar. Intrusion Detection Using Variable-Length Audit Trail Patterns[C]. International Workshop on Recent Advances in Intrusion Detection, Springer Berlin Heidelberg, 2000, 110-129
27. R. Sekar, M. Bendre, D. Dhurjati, et al. A fast automaton-based method for detecting anomalous program behaviors[C]. Proceedings of the 2001 IEEE Symposium on Security and Privacy, IEEE, 2001, 144-155
28. I. Egwutuoha, D. Levy, B. Selic, et al. A survey of fault tolerance mechanisms and checkpoint/restart implementations for high performance computing systems[J]. The Journal of Supercomputing, 2013, 65(3): 1302-1326
29. J. Plank, M. Beck, G. Kingsley. Libckpt: Transparent Checkpointing under Unix[C]. USENIX Winter 1995 Technical Conference, USENIX, 1995
30. J. Duell, The Design and Implementation of Berkeley Lab’s Linux Checkpoint/Restart[J]. Lawrence Berkeley National Laboratory, 2005
31. Boost Documentation. Basic Boost.Asio Anatomy-1.61.0[EB/OL]. http://www.boost.org/doc/ libs/1\_61\_0/doc/html/boost\_asio/overview/core/basics.html, Boost Organization, 2015
32. Linux man page. proc(5): process info pseudo-file system[EB/OL]. http://www.man7.org/linux /man-pages/man5/proc.5.html, man7 Organization, 2016
33. V. Stinner. python-ptrace documentation[EB/OL]. http://python-ptrace.readthedocs.io/en/latest, Read the Docs Organization, 2014
34. T. Carstens, G. Harris. Programming with pcap[EB/OL]. http://www.tcpdump.org/pcap.html, Tcpdump Organization, 2002
35. T. N. Phyu. Survey of Classification Techniques in Data Mining[C]. Proceedings of the International MultiConference of Engineers and Computer Scientists 2009(IMECS’09), International Association of Engineers, 2009, (1): 18-20
36. M. Sujatha, S. Prabhakar, G. L. Devi. A Survey of Classification Techniques in Data Mining[J]. International Journal of Innovations in Engineering and Technology, 2013, (2): 86-92
37. J. Han, M. Kamber, J. Pei. 数据挖掘概念与技术(第3版)[M]. 范明, 孟小峰译. 机械工业出版社, 2012年7月, 226-230

# 攻硕期间取得的研究成果

[A1] 费高雷，杨杰，马路遥，胡光岷。申请专利：基于相似文本反馈的社交媒体文本查询扩展方法。

[A2] ，2016年。

[A3] 电子科技大学研究生二等奖学金，2017年。

## 攻读硕士学位期间参加的科研项目

[1] 全球社交网络分析系统。2016年到2017年。