北京航空航天大學
BEIHANG UNIVERSITY

第二十六届“冯如杯”学生学术科技作品竞赛
论文

科技资源大数据挖掘与可视化

二〇一六年四月

摘要

当前大数据分析挖掘一直是信息领域的热点与焦点，科技领域的应用也在不断进行发展，而包括学生择校择方向、科研人员寻求合作伙伴、对科研人员科研机构的评价与预测等科研领域方面的问题长久以来都未能具有一个正式的平台来进行分析与展示。为此我们爬取了千万数量级的论文、专利及项目信息等科研相关信息，并对这些数据从人物、机构、领域等多个方面进行深入分析与挖掘，并将挖掘结果以可视化的形式展示出来，并以搜索引擎的形式使用户进行自定义检索以获取其所关注的信息。目的在于建立多用户、多维化、多分辨率、多种类的科研领域数据体系，深入挖掘和应用科技信息资源，加强科研行业数据资源的综合化、共享化并提高利用率，为国家科研领域信息化的规划、建设和管理服务，为学生、科研人员、科研机构、政府部门等提供科研服务。

本文根据上述情况，首先介绍了科技资源搜索引擎的建立背景，然后从系统概要设计的角度对系统进行了详尽的整体框架与设计方案上的介绍，进而介绍了引擎建设过程中所涉及到的关键技术与创新点，然后对本引擎当前成果做出了完整的应用功能展示，最后详细说明了项目的实现意义与市场价值收益及应用前景等问题。

关键词：大数据，科技资源，搜索引擎，数据挖掘

Abstract

Big data analysis and mining have been currently a hot spot and focus in the field of information. Although the applications for scientific research have been continually developed these years, the problems including how students choose their schools and majors, how researchers search for their partners for cooperation and how to evaluate the academic aptitude of so many researchers and their institutions have never been solved satisfyingly and there is not a formal platform to analyze and display. To handle this task, we have crawled thousands of orders of magnitude of scientific research information, such as academic papers, patents and project information, and the data from different fields of people and institutions have been analyzed and mined in multiple aspects with the mining results being displayed in the visual form. This platform uses the form of search engine so that users can customize the retrieval in order to get the information they want to pay attention to. Our aim is to establish a multi-user, multidimensional, multi-resolution scientific research system with various kinds of fields taking into account, to deeply mine the data and apply the science and technology information resources, and to provide services for the planning, construction and management of the national scientific research field and to provide scientific research service for related scientific research personnel.

Based on above-mentioned circumstance, this paper will firstly introduce the background of the establishment of the science and technology resources search engine, and then we demonstrate the system structure and design scheme from the aspect of system summary design and overall framework in detail, where the key technologies and innovative points of this system will be introduced in the process of the construction of the engine system. Then, the current achievements of this search engine system will be completely displayed with all the functions. Lastly, the significance, market value and application prospects of this search system will be illustrated in detail.

Keywords: Big data, Science and technology resources, Search engine, Data mining

目录

图表清单

一、引言

（一）项目背景及需求

大数据挖掘技术在当今是时代的热点与焦点，越来越多的领域也将大数据的挖掘作为研究本领域结构、趋势及动态的重要参考。而科研领域在大数据使用上却依然存在空白，并且存在用户缺乏途径去获取需求科研信息，以及信息源间信息无法互通的信息孤岛现象，利用这些信息进行数据挖掘更是难以实现。

（二）研究内容

针对上面所提到的需求，我们在万方、百科立方、中国知网、国家自然基金网等科技科研网站平台爬取了共计 12195852 篇论文、8855990 项专利、458176 个重大科研项目信息、14719 台大型科研设备信息。覆盖了 743607 位科研人员、177 所科研机构、约 25 万大小领域的信息数据、关系网络及动态趋势。并利用这些信息数据使用 Solr 做索引存储，充分挖掘人物、机构、领域关系网络及动态趋势，最终以搜索引擎的形式对关注内容检索，并将挖掘结果以可视化形式展示。

（三）国内外现状

1、国内现状

国内在科技资源分析、评价与预测方面仅停留在包括人物、机构信息的简单罗列及介绍，在领域方面也仅针对国家明确的一级学科、二级学科来进行机构排名，如武书连的大学排名、清华的 Arnet Miner。而尚未具有基于海量数据条件下的自定义搜索途径，并不具有本项目的关联分析与挖掘技术，更不具有如本项目的简明扼要的展示手段与丰富的搜索范围。

2、国外现状

国外也尚未具有有显著影响力的针对科技资源数据挖掘分析评价的透明化平台，对机构评价也仅停留在数据对比及排名上，而未利用拥有的数据进行挖掘分析处理。

二、技术分析

（一）爬虫程序

爬虫程序是指通过按照用户给定的规则来自行爬取 Web 信息的程序。关于爬虫具体的工作流程简述：首先要依据 Web 自适应分析算法筛选出不相关链接，同时将有效链接放入链接队列等待捕捉。然后将依据给定的搜索算法从队列中选择 Web 链接作为下一步捕捉的入口，之后就是不断进行进队出队操作到满足预设值为止。在捕捉过程中，被抓取的网页还会不断被存储、分析、筛选、建立起索引结构，以便于数据的查询检索。现如今，网络上成型的爬虫程序已多达上百种，而由于科技资源信息的特殊需求和 Solr 接口的约束，本项目选择了 Nutch 作为爬取数据的工具。而在爬取过程中遇到的特殊信息，则使用了自己编写匹配规则和逻辑的爬虫程序采集信息。

Nutch 由爬虫程序 crawler 和查询组件 searcher 组成。Crawler 主要作用在于抓取 Web 页面并将 Web 分析过滤、建立索引。而 Searcher 主要则是利用建立的索引来对 Key Word 进行检索，将产生的结果进行存储的功能程序。由于平台的特殊性，我们仅使用了其 crawler 部分，因此在此仅对 crawler 部分进行详细的分析说明。

关于 Crawler，重点分析其工作流程和涉及的数据文件。数据文件主要包括 web DB、segment、index 三类。Web DB，也叫 Web database，存储了爬虫捕捉的 Web 间链路结构信息。在捕捉过程中会收集到多个 segment，每个 segment 中存储有 Crawler 一次捕捉中捕获到的 Web 及其索引。Index 是捕获的所有 Web 的索引，是所有 segment 中的索引的集合。

Crawler 的执行流程：Crawler 首先依据 Web DB 文件产生待捕获 Web 链接集合 Fetch list，然后下载线程 Fetcher 依据 Fetch list 捕获 Web，若 Fetcher 有多个，则生成多个 Fetch list，即一个 Fetcher 对应一个 Fetch list。Crawler 用捕获得到的 Web 更新 Web DB，再依据更新完的 Web DB 产生新的 Fetch list，其中包括未捕获的或者新产生的链接，最后下一次的抓取进程开始进行。形成“生成-捕获-更新”的循环。

Crawler 程序由以下子程序组成。这些程序在 Nutch 有相应的命令行来执行单独调用。下面就是这些子程序的功能描述。

1. 首先创建新的 Web DB (admin dB -create).
2. 将抓取起始链接写入 Web DB 中 (inject).

3. 将 Web DB 生成 fetch list 并写入 segment(generate).
4. 依据 fetch list 的链接抓取 Web (fetch).
5. 依据抓取 Web 更新 Web DB (update db).
6. 循环进行若干步至设定深度。
7. 依据 Web DB 得到的 Web 评估链接，同时进行 segments 的更新 (update segs).
8. 对捕获到的 Web 进行索引(index).
9. 将索引中重复的 Web 和重复的链接除去 (dedup).
10. 将 segments 中的索引合并并生成最终的 index 进行检索(merge).

（二）Solr

Solr 是 Apache 的基于 Lucene 的企业级 search engine 服务器的开源项目。它对外提供 Web-service 的 API 接口。需求者可通过 http 请求，向 search engine 服务器提交固定格式的 XML 生成索引。也可利用 Http Get 提出查找请求，返回一个 XML 结果。

本项目数据的结构化存储便是依托于 Solr 进行索引查询。

（三）相似性计算

本项目涉及很多计算相关性的问题，例如，人物之间的相关关系和领域之间的相似性、对检索结果进行排序。最简单的实现方法是关键词匹配，例如传统的搜索引擎。再进一步考虑，可以使用基于语义的相似度匹配。其核心技术在于实体的特征向量表示和向量集间的相近程度。针对不同的应用和数据大小情况、时间空间花费等限制，相似度计算方法的选择又会有所区别。

关于详细的计算方式，我们采用了余弦相似度的计算：余弦相似度，原理是若两句话用词越相似，那么内容会越相似。因而可从词频来进行相似性计算。比如这里有图 1 两个句子进行分词后的结果。

句子A：我/喜欢/看/电视，不/喜欢/看/电影。

句子B：我/不/喜欢/看/电视，也/不/喜欢/看/电影。

图 1　句子切分

那么我们现在有“我、喜欢、看、电视、电影、不、也”七个词，依据其在两个句子中出现的频率可以得到他们的词频，进而按这七个词的顺序可以将两个句子表示为两个七维向量如图 2 所示：

```
句子A：[1, 2, 2, 1, 1, 1, 0]

句子B：[1, 2, 2, 1, 1, 2, 1]
```

图 2　句子向量

之后通过如图 3 所示的余弦相似度公式来计算其夹角的余弦值，余弦值越趋近于 1，表示两个向量的方向越统一，相似度也应越高。

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

图 3　余弦相似度公式

我们还采用了随机超平面 hash 算法 simhash 和海明距离的计算：仍以上面两个句子为例，首先对句子中每个切分出的词依据其出现频率及词性来确定其权重比如“我”取为 2、“喜欢”取为 4、“电视”取为 5，之后通过哈希算法把每个词变为对应的哈希值，如“我”hash 为 100101，“电视”hash 为 101011，之后将之前的权重加入其中，如“我”hash 值为 100101，权值为 2，则加权后为 2 -2 -2 2 -2 2，同样，“电视”为 5 -5 5 -5 5 5，其余词语也做类似变换，之后对每个句子将其所有词语对应位置的权值相加，比如“我”和“电视”相加得到 7 -7 3 -3 3 7，其余类似，将本句的所有词语相加得到的结果，按大于 0 记为 1，小于 0 记为 0 的规则做降维变换，如 7 -7 3 -3 3 7 变为 101011。之后依据降维结果对上面两个句子进行海明距离的计算。两个句子海明距离即为两个 simhash 对应二进制取值不同的数量。如 10101 和 00110 的海明距离为 3。由此我们取得了句子间的相似程度关系，由此可以得到检索结果的权值排序等结果性信息。（样例中的 hash 结果和权值赋予是为了使算法举例清晰易懂而取的虚拟值）

（四）人名和机构名称抽取和规范化

原始的数据源上，无法直接获得作者与作者单位之间的对应信息，因此只能先获取作者的拼音形式姓名与英文机构名称之间的对应关系，然后进一步获得汉字形式的姓名与中文机构名称之间的对应关系。

（五）人名消歧

首先，为了避免分词不当造成机构名称字符串本身粒度被破坏以及过度概率化建模导致问题复杂化，弃用 VSM 模型表示文本的策略，而将单位字符串本身看作一个整体；其次，在聚类算法上采用不需要事先估计类别数目的、基于图论的聚类算法，之所以采用这一算法，另一层面的考虑是由于图具有关联推断的特性；再次，从字符串的粒度上，采用最长公共子序列（Longest Common Subsequence, LCS）以及最长非对称前缀(Longest Asymmetric Prefix, LAP)作为特征进行相似度计算。同名作者消歧算法，以就职机构为主要区分特征，无法识别同机构人物的同名问题，当同一个人物更换工作单位且从题录数据中无法挖掘到履历变迁时，或者人物 的工作单位名称发生较大的变化时，会将同一个作者识别成多个作者。以上两个问题，考虑在现有算法框架内融入更多的特征，如学者共发文关系，同机构关系，论文发表刊物，论文发表时间等进一步改进。

主要有三条加边规则。如果两个结点的单位字符串属性的最长公共子序列的长度与其中较小字符串长度之比大于一定阈值 25（称之为主规则），并且两个结点的单位字符串属性之间的最长非对称式前缀大于 2（称之为辅助规则），则在两个结点之间加上一条边，本条规则依靠主规则计算单位字符串之间的相似性，借助辅助规则来避免主规则将类似于“华中理工大学”与“华南理工大学”的单位字符串归为相似的情况；如果第一条规则失效，接着检测两个结点的单位字符串的最长非对称式前缀中是否包含 Rule Main 中的模板，如果包含则在两个结点之间加上一条边；如果前两条规则都失效，则接着检测两个结点的单位字符串的最长非对称式前缀只能是否包含 Rule Supplement 模板，如果包含则在两个结点之间加上一条边。

基于图的连通分量的人名消歧算法，在构造的图上，每个连通分量分别对应于现实世界的一个作者实体，求出图上的所有连通分量，也即获得了实体指称项与实体之间的对应关系。本文以图的深度优先遍历算法为基础设计的求无向图上所有连通分量算法。

（六）数据展示

随着大数据技术的发展,数据的展示也日益成为一个研究的热点。大数据展示技术主要关注如何将数据计算的结果以一种形象的、直观的图表来展示经由数据分析得来的结果。

本项目中包括关系图、趋势图、分布图等的数据展示主要使用 Echarts 及百度地图热力图 API 来进行展示。Echarts 是一个基于 Java Script 的商业级数据图表库，能便捷、灵活地在 PC 端和移动端进行数据的展示，同时能兼容 Firefox、Chrome、IE6.0+等当前绝大部分主流浏览器，底层依赖 Canvas 的 Zrender 类库，具有鲜明、交互性高、可个性化的大数据可视化图表。同时 ECharts3.0 的版本还具有拖拽重计算、数据视图、值域漫游等特性，使得用户体验效果增强，同时给予用户对数据进行自由挖掘、整合的能力。

具体展示功能会在第四章功能展示中得以体现。

三、实现方案

（一）爬虫程序设计

爬虫程序结构如图 4 所示。

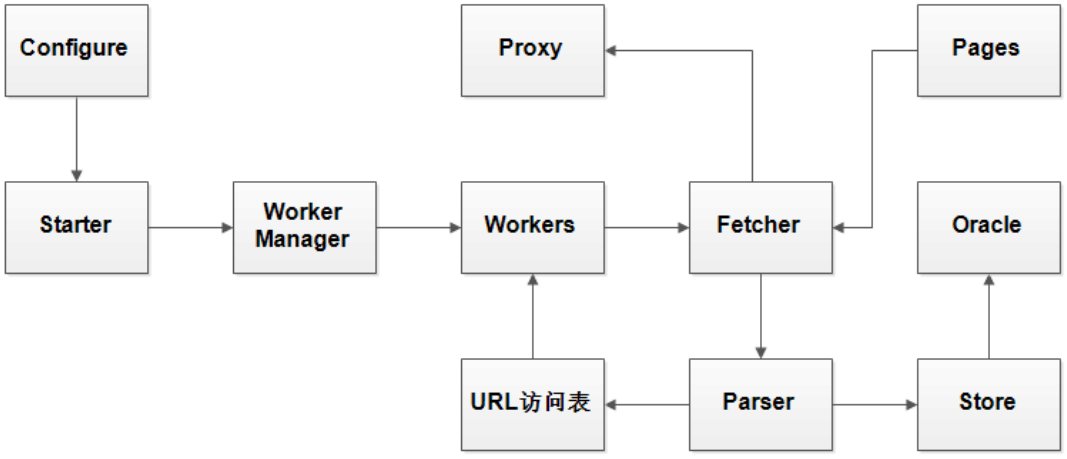

图 4　爬虫程序结构

1.Configure: 主要职责为存储给定的初始捕获链接地址和配置信息。

2.Starter: 职责为读取配置信息、启动抓捕 catch 程序等程序初始化的工作。

3.Worker Manager: 主要起到运维控制 Workers 线程的作用，并且能够监听其数量及状态。

4.Workers: 通过利用 Configure 缺省表的线程配置的执行线程数来进行具体的网页信息捕捉存储解析工作。

5.Fetcher: 主要负责进行 Web-require 和 Web 存储下载工作。

6.Proxy: 主要起到运维控制向上一线程提供的 IP 代理服务的作用。

7.Parser: 主要通过开源 Web 解析存储工具 JSOUP 来对 Web 进行解析存储，并依据实际情况配置 Web 解析存储工具。

8.URL 访问表: 主要起到运维 Visited 和 Not-visited 链接列表的作用，链接列表通过一种以关键阈值为核心的 Radis-memory 分布式数据库实现，同时结果会传递存储至 Store。链接访问列表又分为 Visited 和 Not-visited 两种。

（二）Solr 结构设计

Solr 搜索引擎的架构图如图 5 所示。

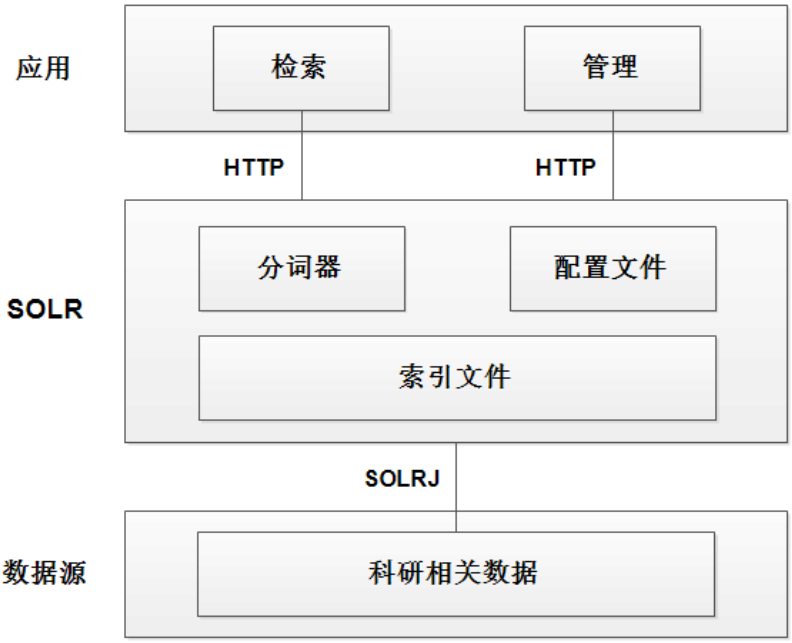

图 5 SOLR 结构

最底层为所爬取捕捉的科研数据，在 Solr 服务器中进行结构化存储，提供 Solrj 索引接口给上层。

第二层是 Solr 的核心，由一些索引、配置文件和 Solr 的分词器组成。主体是基于 servlet 的实现，因此要将 Solr 发布在 Vector 服务器上运行，本项目中使用 tomcat 服务器，将 Solr 的文件置于 tomcat 的 web roots 目录下运行 tomcat 便可以使用 Solr。

顶部为应用层部分，Solr 自带的 HTTP 管理端功能全面，可进行包括索引查询、管理等事务处理。查询也可利用 Solr 内部的 HTTP 访问 api 进行索引检索。

Solr 索引的目的主要是为了提高数据查询效率，所以在 Solr 建立了包括论文、专利、项目信息、人员、机构、关系网络信息等 10 项索引表。同时，Solr 采用了倒排结构来索引文档，倒排索引结构如图 6 所示。

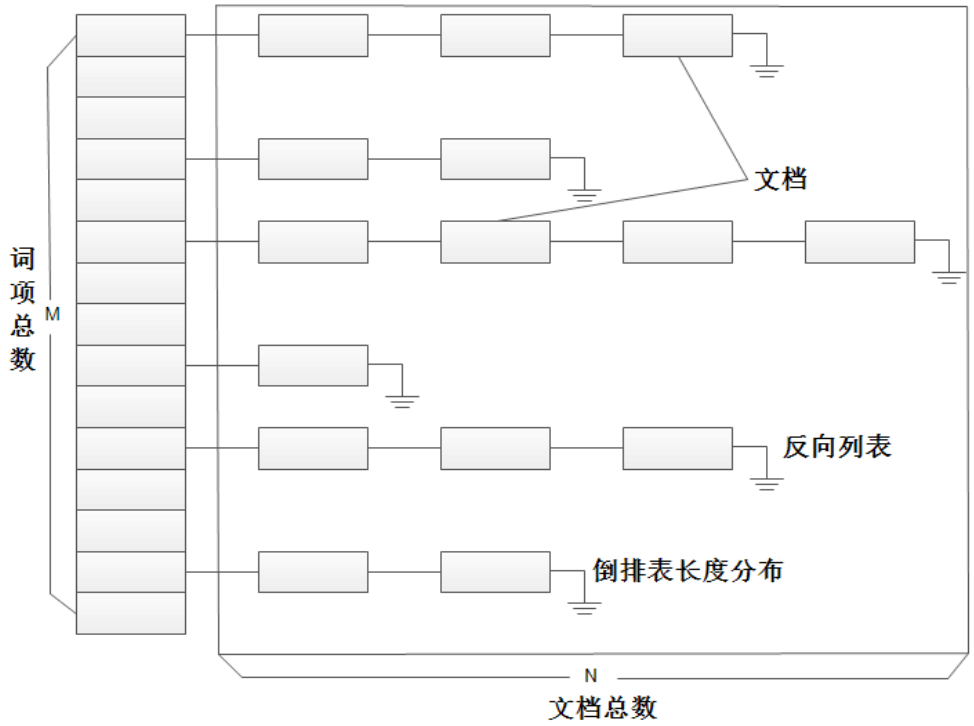

图 6　倒排索引结构

（三）平台构架设计

1、总体构架

科技搜索采用了如图 7 所示的四层构架，分为展示层、分析层、处理层和数据层。数据层包括了通过爬虫捕获和各种类型的原始数据，处理层对数据层的数据进行抽取、清理和统一，分析层对统一后的数据进行各种分析以便交给展示层进行展示。展示层直接处理用户的检索需求，对用户的检索请求进行响应返回对应的分类结果。

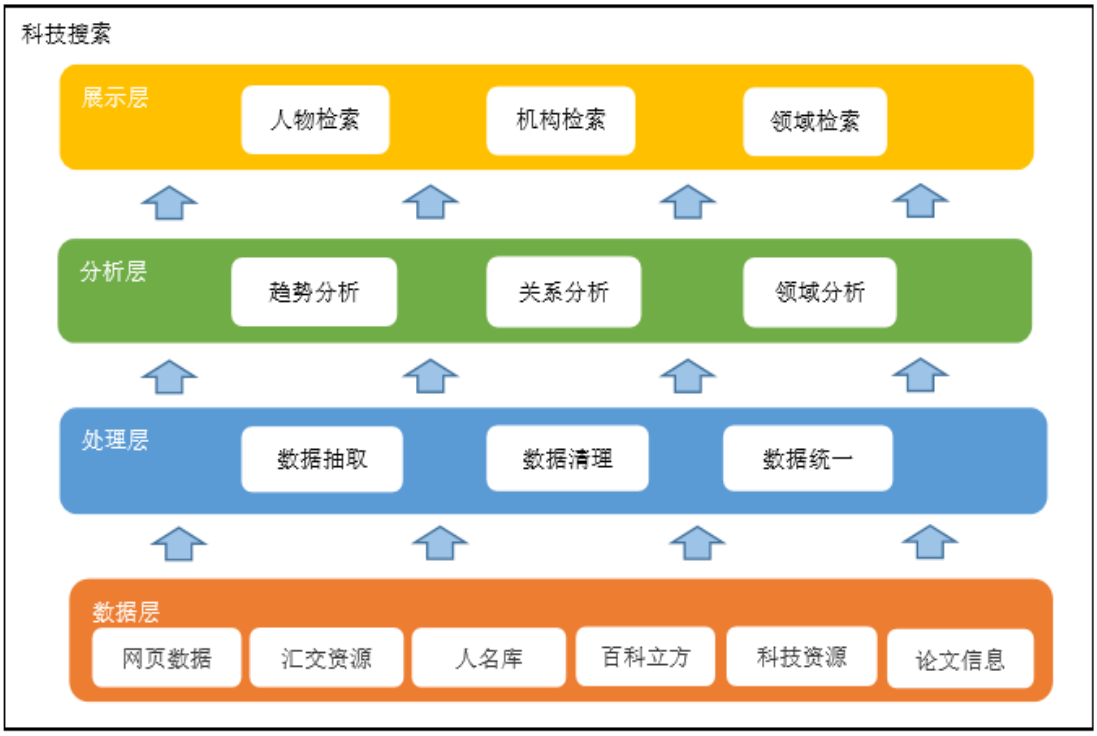

图 7　平台总体构架

2、数据层

数据层中包含了使用爬虫爬取的科技资讯数据、汇交课题和自然基金项目中的元数据、来自人名库的元数据信息、百科立方项目中所存储的词条关系数据、来自共享网的科技资源数据、爬取的论文库信息等。这些众多的数据来源格式差异比较大，需要经过进一步的处理才能够进行分析和检索。

3、处理层

对于数据层所提供的数据格式不一，处理层对其进行抽取、清理和统一，将数据格式统一为 Lucene 索引格式，便于下一步的处理。同时也能够从元数据中提取出更多的信息，如从论文和项目信息中提取出人员和机构的信息供下一步的分析。

4、分析层

在将不同来源的异构数据进行统一以后，就可以进行下一步的分析。如对于人员、机构数据都可以根据其项目和论文情况对其科研热度趋势进行判断，对于领域类别也可以使用论文和项目数据来对某领域的热度进行分析。对于人物、机构、领域之间的关系

也需要使用百科立方的数据来进行分析和提取。对于领域来说，需要对领域内知名的人物和知名机构进行分析和挖掘。

5、展示层

在分析层进行分析以后，根据用户提交的检索请求，按照人物、机构、领域三个大类进行结果的展示，每个大类又对应许多的小类信息，如与人物相关的小类信息有人物详细介绍、人物关系图、论文信息、相关资讯、相关项目等。用户也可以进一步查看与之相关的详细小类信息，达到对有关人物、项目或领域的深入了解。

四、功能实现

（一）使用说明

使用用户针对科技人物、机构、领域三方面通过键入自己想要了解的关键字进行检索以获得相应方向的挖掘成果、趋势分析与结论信息。

（二）部分成果展示与分析

1、检索页面

检索页面如图 8 所示。

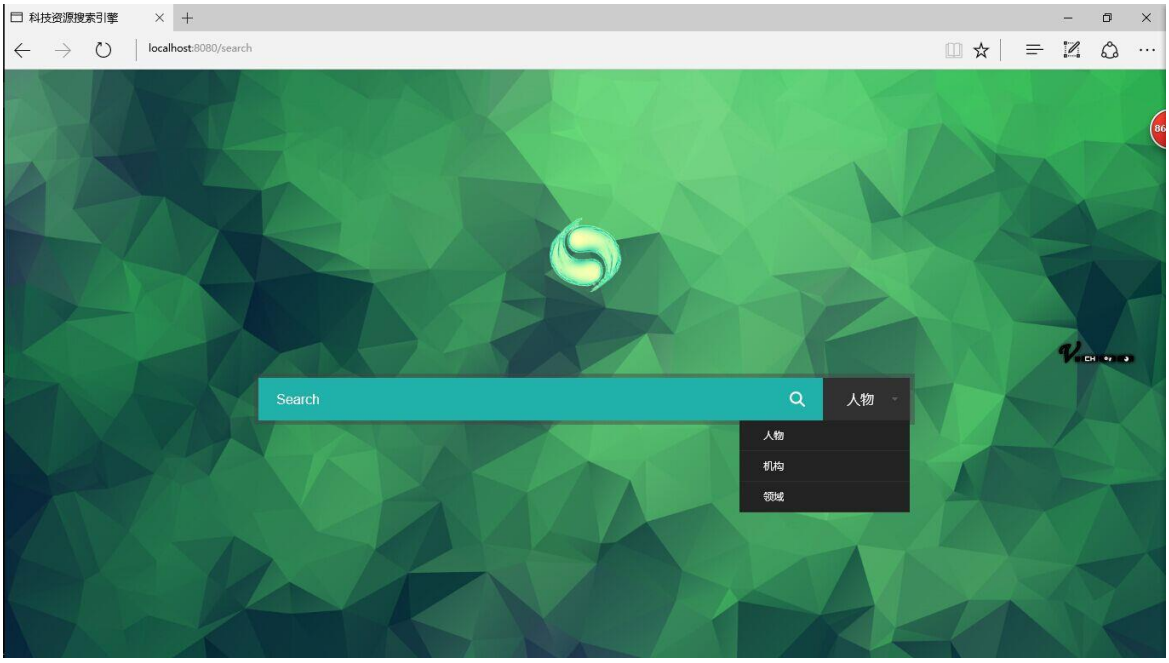

图 8　检索页面

为一个常规的搜索引擎界面，用户在检索框体键入需要检索信息，在搜索框右侧选取查询方向：科技人物、机构或领域进行查询检索。接下来对三个方向通过检索具体信息来展示部分重要功能，因篇幅原因未能展示的部分功能会在附加视频中体现。

2、科技人物检索

下面通过检索前北航校长“怀进鹏”怀院士为例来对科技人物方向检索功能做出说明：

如图 9 所示为所检索人物的研究领域及方向上的挖掘信息，通过云标签的形式进行呈现，其中由于领域间所发表数量与质量的不同，依权重来确定标签大小与布局。依据挖掘结果可清晰掌握所关注科研人员的研究方向，能满足如学生在导师选择、科研人员在课题寻求合作伙伴等需要。

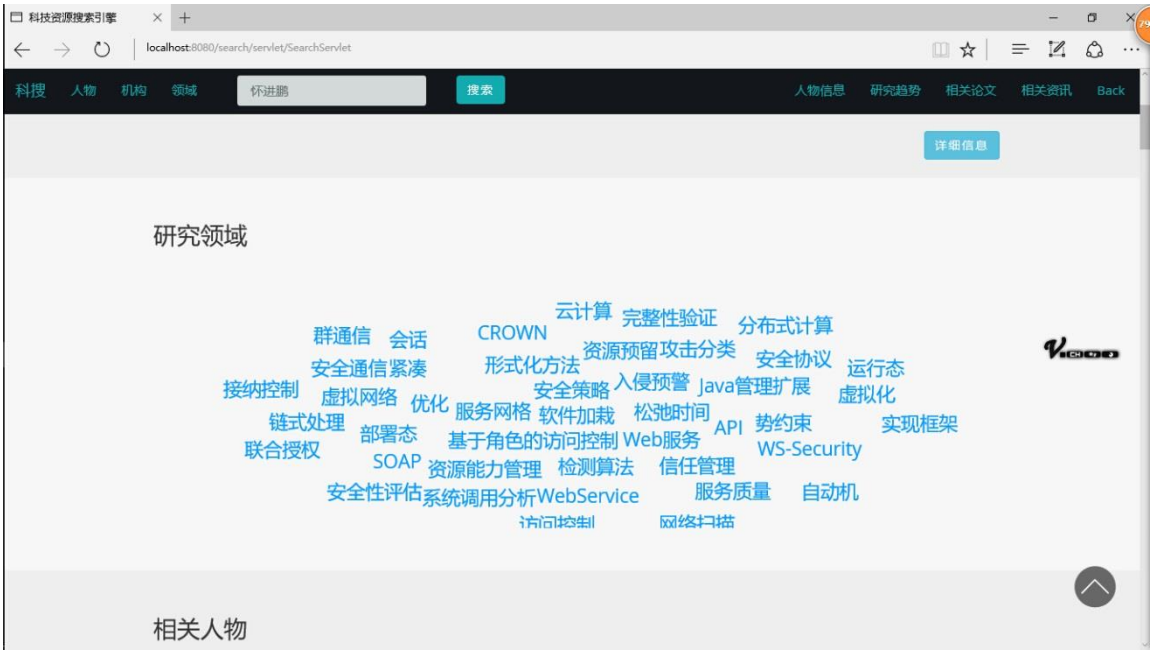

图 9　科技人物研究领域

如图 10 所示为挖掘得到的所检索人物的科研关系网络，同时网络节点依据不同关系划分了不同颜色，同时具体的论文合作项目也在边上注明。依据此项挖掘成果可清晰了解检索人物的科研人际资源及影响力与重要程度。

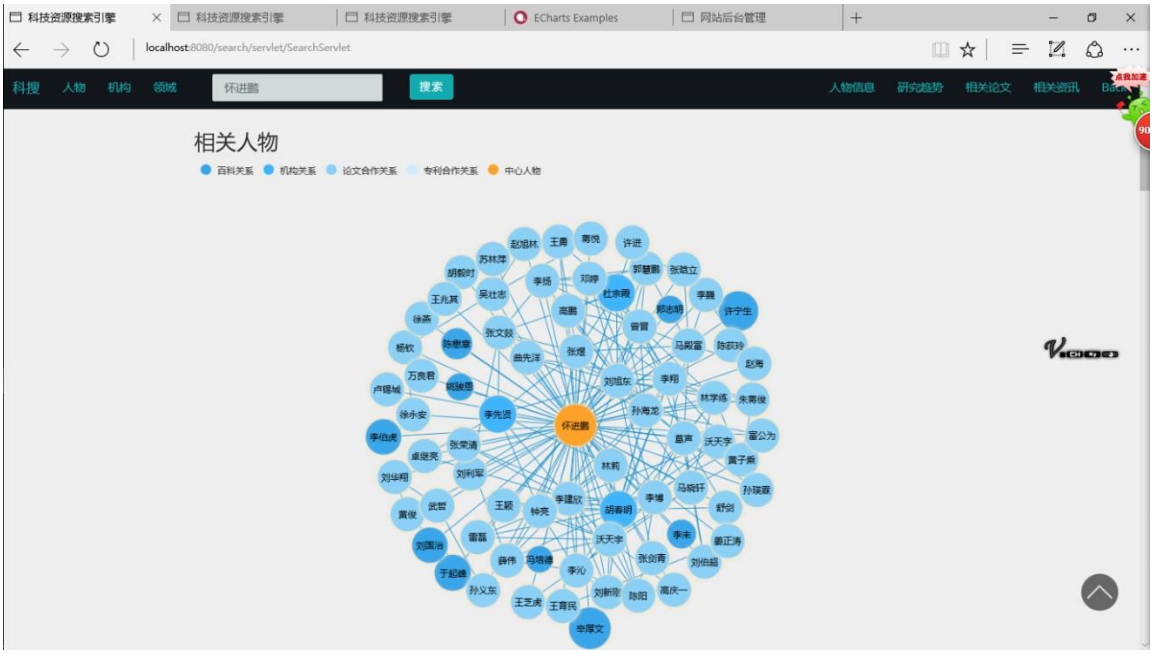

图 10　科技人物关系网络图

如图 11 所示为所检索人物近 10-20 年的科研活动趋势，分为论文、专利、项目三个条目，利用折线图清晰地了解人物的趋势与动态。便与对其形成准确的评价与预测。

图 11　科技人物研究趋势图

如图 12 所示为所检索人物近 10 年的项目、论文质量分析雷达图，对于论文作品分别从数量、下载量、被引量、导师能力、期刊水平、一作指数六个方面进行评估，而对于项目则通过结题成果、参与影响、项目影响、项目规模、领域热度五个方面进行评估。是一项综合化的挖掘成果。

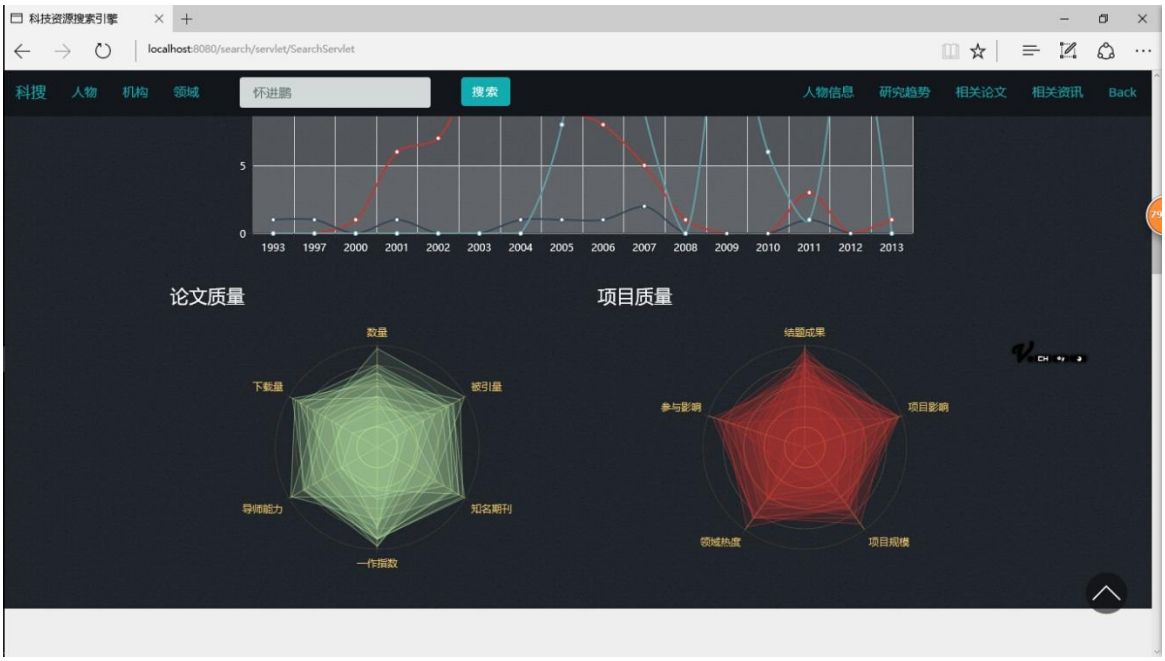

图 12　论文项目质量评估

3、科研机构检索

下面通过检索北京航空航天大学简称“北航”为例来对科研机构方向检索功能做出说明：

如图 13 所示为所检索机构的简介内容及领域影响力，依据本机构人员的论文水平及科研方向与影响力作出权重排名，并将其领域影响力以字符云标签大小呈现出。该挖掘结果清晰有效地评价了科研机构的研究方向与领域影响程度。对包括学生择校、相关部门进行科研评估有重要参考价值。

图 13　科研机构基本信息图

如图 14 所示为该科研机构的人员构成，并依据科研人员的职位及影响力重要程度来对节点大小做加权处理。挖掘结果清晰有效地展示了机构的结构并可据此作出人物影响力鉴别。

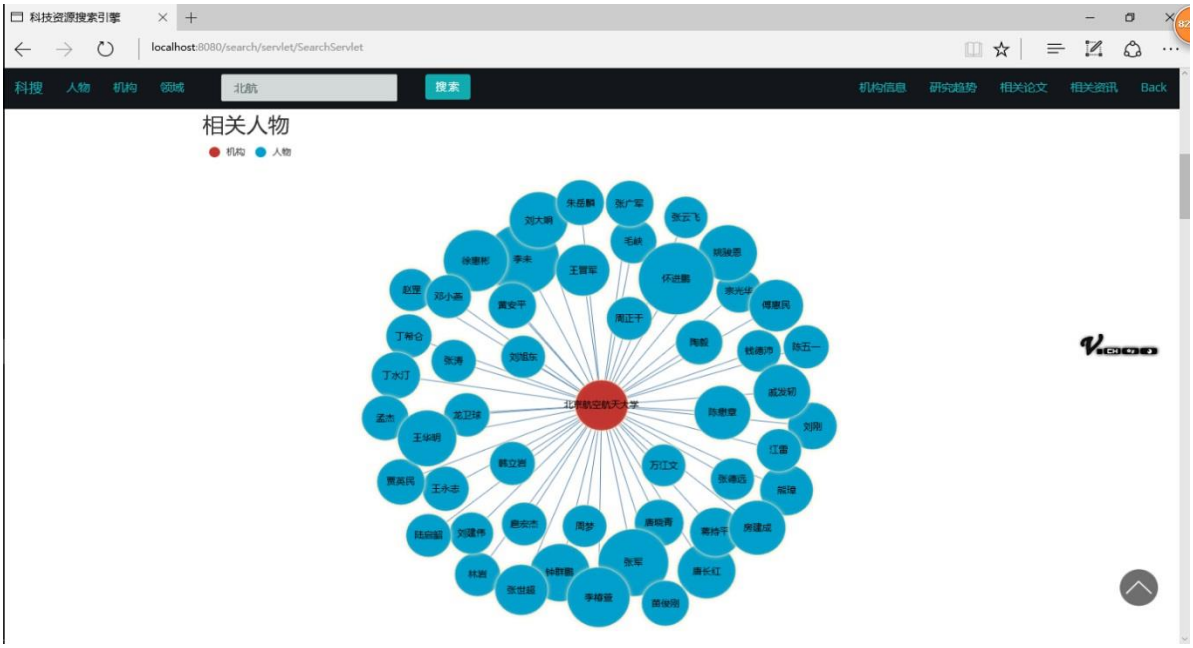

图 14　科研机构人员构成图

如图 15 所示为所检索机构近 10-20 年的科研活动趋势，分为论文、项目两个个条目，利用折线图清晰地了解该科研机构的趋势与动态。便与对其水平与能力形成准确的评价与预测。

图 15　科研机构活动趋势图

4、科技领域检索

下面通过检索当前热门领域“计算机”为例来对科技领域方向检索功能做出说明：

如图 16 所示为该领域的邻域关系网络，颜色区分了其研究方向，大小显示了其当前的热门程度，该方面的挖掘，使用户对检索领域邻域作出进一步认知与分析，方便其进一步查询相关领域内容。

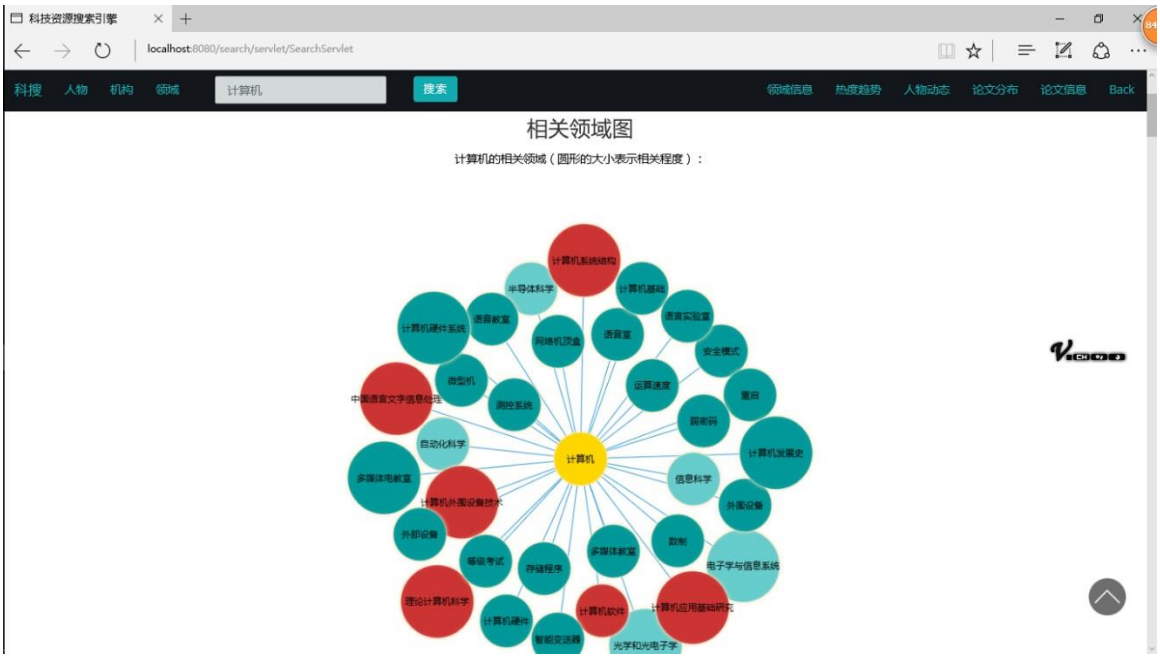

图 16　科研领域邻域关系网络

如图 17 所示为该领域范围内的科研机构与人物的贡献度排名，依据机构性质、人物职称、论文专利项目参与程度、研究领域的趋势与热度进行加权综合排名所得到的综合结果。在本项目中具有重要地位。对学生、科研人员、机构及更多相关科研组织认识、掌握领域动态及领域内人员地位、重要程度有极为重要的指导意义。

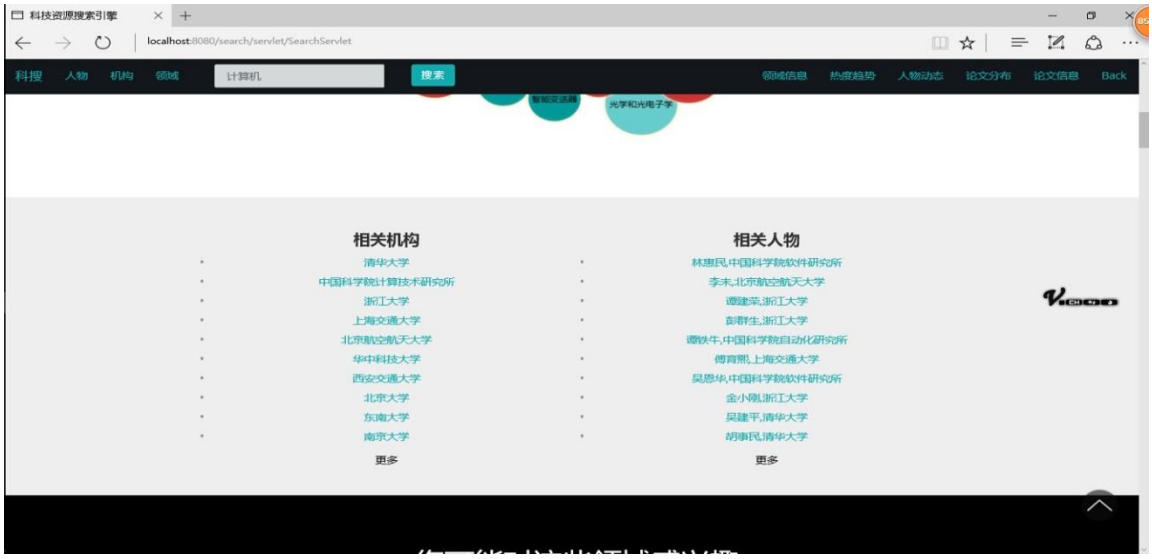

图 17　科研领域相关人物机构排名

如图 18 所示为所检索领域的邻域标签云，依据其热度控制标签大小，在搜索方面细化用户需求，起到二次检索的作用。

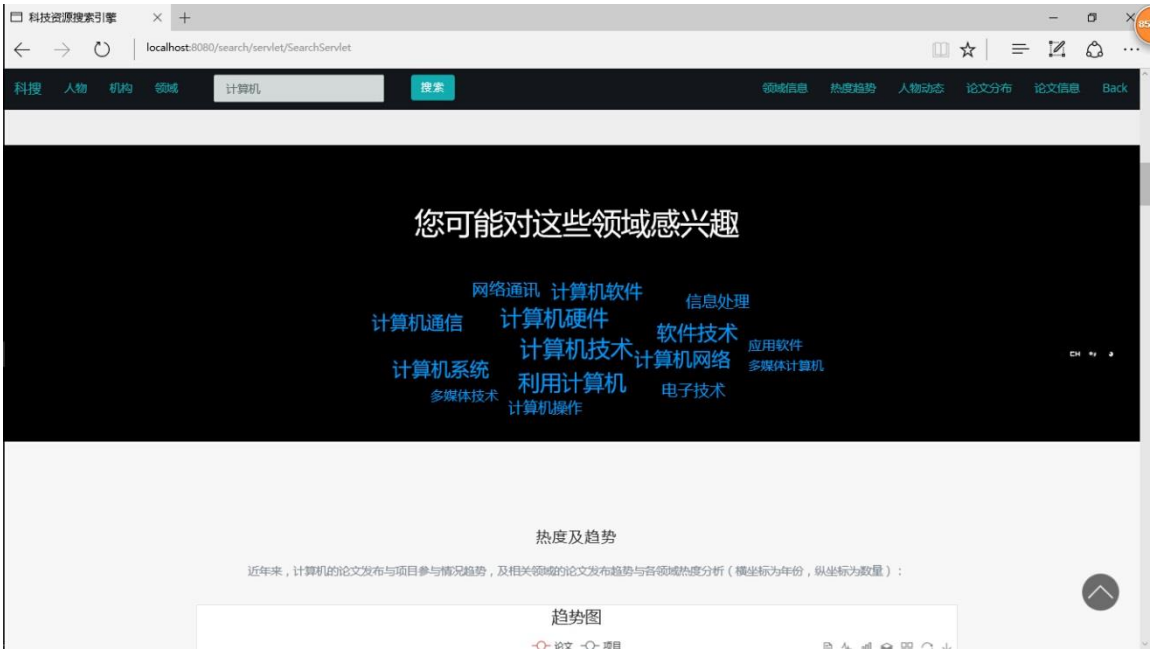

图 18　科研领域邻域标签云

图 19、20、21 三张图分别为领域的论文项目活跃趋势、将邻域内容融入扩展后的总趋势动态、领域与其邻域趋势的对比图。三张图从三个方向剖析了检索领域的动态情况，使用户对领域总体发展动向与趋势做出明确的把握。

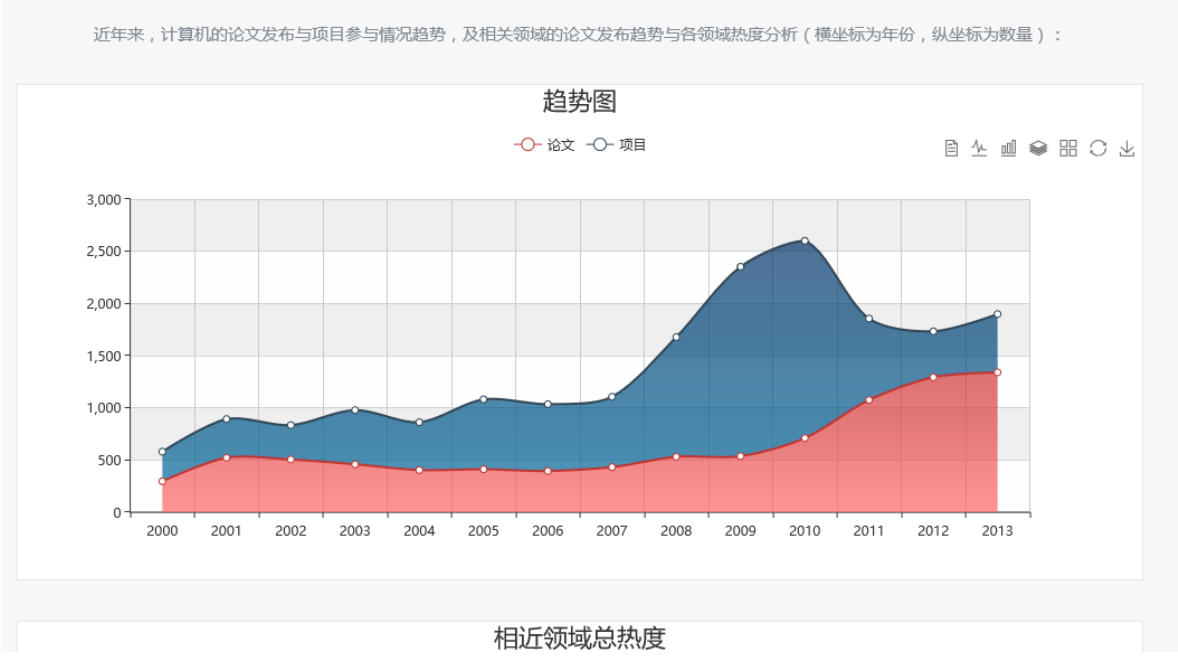

图 19　科研领域活跃趋势图

图 20　科研领域邻域总热度图

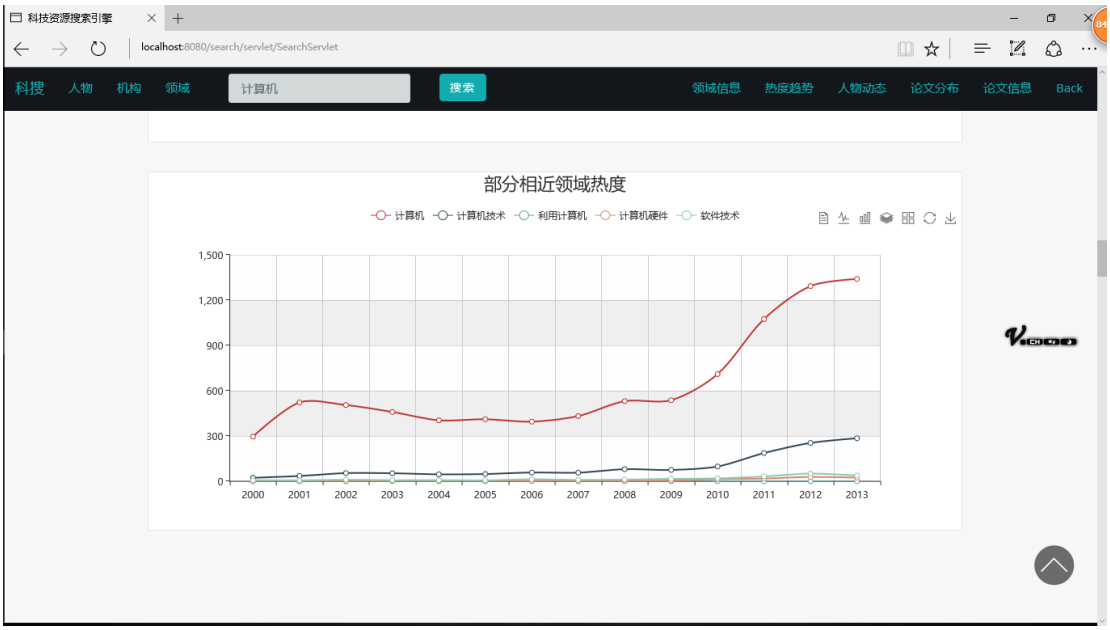

图 21　科研领域邻域对比图

如图 22 所示为所检索领域近 10-20 年重要人物的论文、专利、项目的参与动态展示，横纵坐标分别为人物该年参与的论文与专利成果数量，不同的点代表不同重要人物，鼠标悬停可查看其具体信息，点的大小代表其参与项目的数量。将以上四种元素相融合，以时间轴作为主线来展示其科研动态情况。

图 22　科研领域重点人物动态

如图 23 所示为所检索领域的论文分布情况的热力图形式展示，此处使用了百度地图的 api，并且由于数据的准确性，分布地图可放大查看具体论文发表机构及相关论文信息。

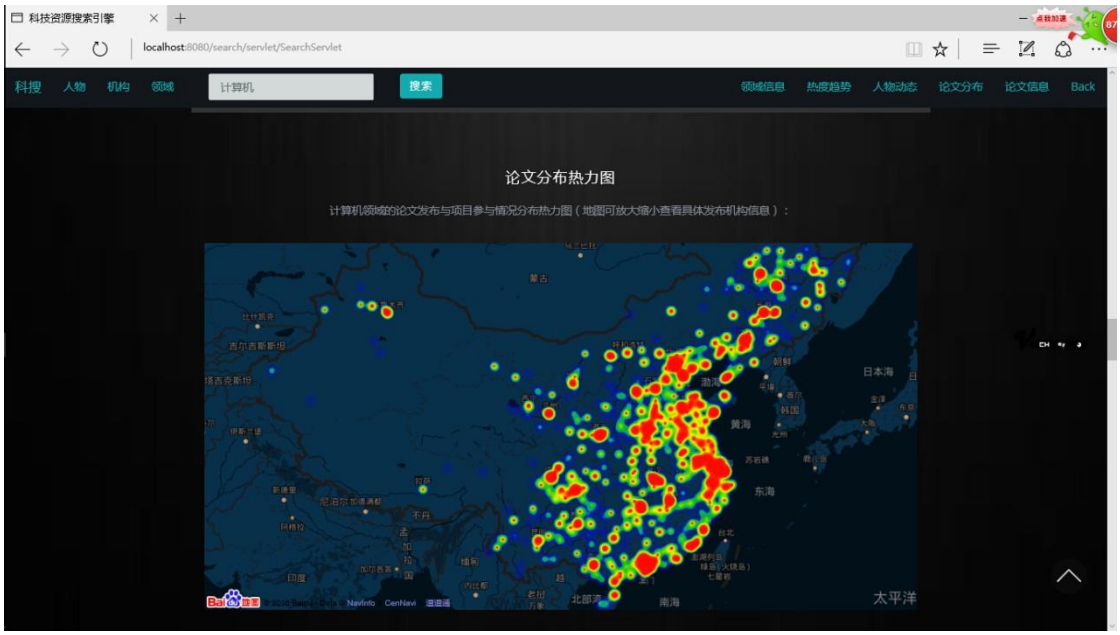

图 23　科研领域论文分布热力图

五、意义与前景

（一）技术特点与优势

1、优势

a.海量、全面的信息数据爬取，达到千万数量级。

b.高效的存储索引结构和搜索算法。

c.自由度高的搜索呈现方式。

d.严谨鲜明的挖掘分析结果和组织方式。

2、创新点

a.以论文、项目、专利等与科研成果密切相关的方面作为切入点进行达千万数量级的海量数据爬取与挖掘来分析评价预测。

b.将挖掘结果以搜索引擎这一自由度极高的形式进行展示。

c.对数据采取云存储架构与多级索引存储结构并使用 Solr 进行海量数据的检索。

d.其他细节处理，如缩写及同义词的模糊处理、同名人物记忆化数据分析、领域识别及二次检索、机构与人物的排名算法及权重处理等。

（二）实际应用价值和现实意义

对国内大多数学者、机构和领域进行详细评价和影响力分析，可在高校学生择校、选择导师、专业选择，研究人员分析行业领域趋势、寻求科研合作伙伴，国家科技部门评价机构及个人、分析领域等多个具体方向有广阔的应用空间。

（三）使用范围、前景及效益预测

可用于包括学生择校择师择方向、科研人员分析研究领域寻求合作伙伴、对科研人员科研机构的评价与预测等多个科研活动中必不可少的环节中。推广主要投入在高校、科研机构中推广。经济收益上可采用使用收费或传统搜索引擎广告收费模式。收入空间团体极为广阔。

（四）项目未来工作

1、数据方面

数据方面目前仅局限于中文文献，日后会加入大量国际期刊论文信息，功能上由于非中文文献的加入将会对搜索与数据处理进行大规模升级，同时分词算法也会进行大幅度的修改。

2、科研人物关系网络

我们还试图基于已有的大数据资源进行专注于科研人物关系网络的开发，如概念图所示。其中每个节点均为对应的科研人物，所有科研人物组成整个星系，而人物又依其研究领域在星系中划分入不同空间，而依据其影响力其节点大小也相应改变。

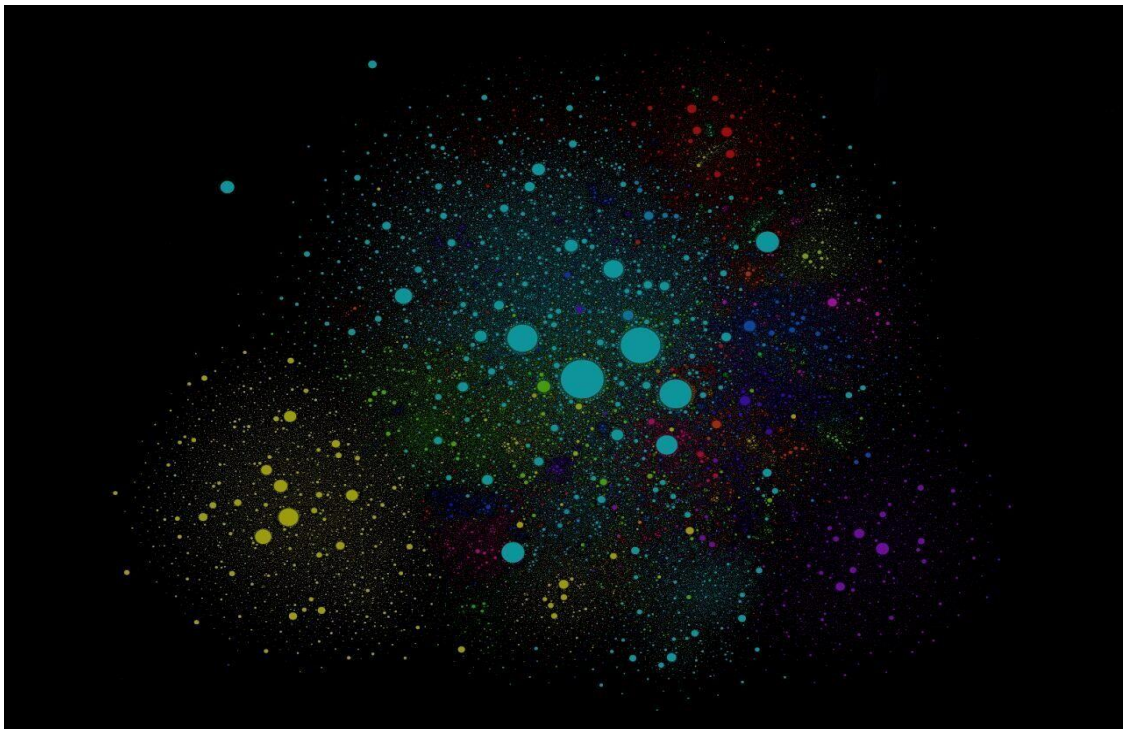

图 24　科研关系网络概念图-总图

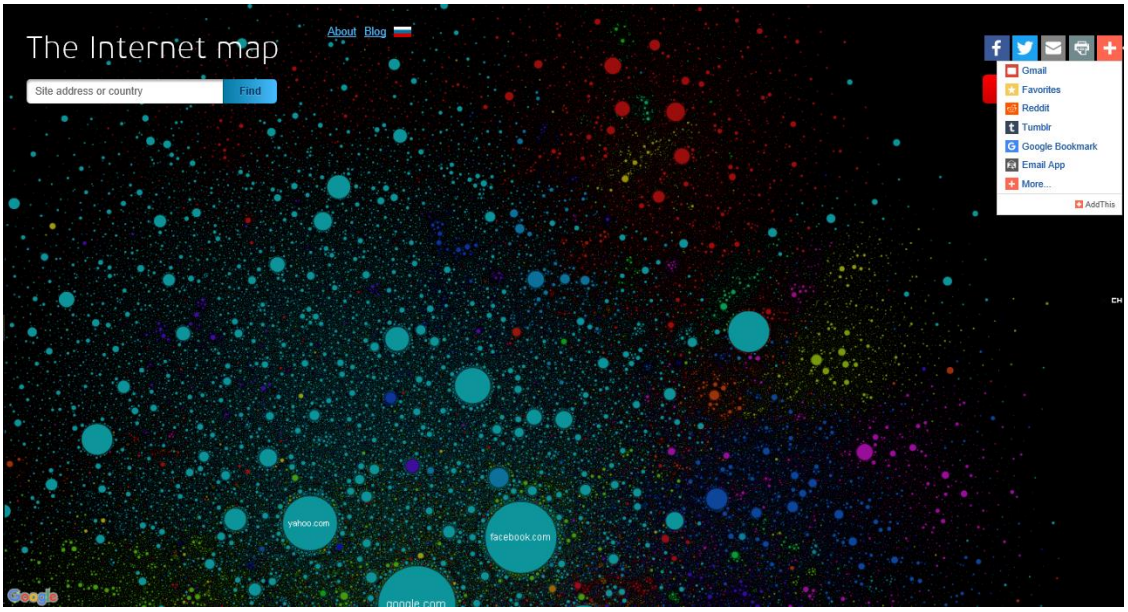

图 25　科研关系网络概念图-局部图

结论

大数据是新时代发展的热点与焦点，通过与各领域各行业的结合，引导着各领域的改革，不断促进其发展与进步。科研领域是大数据应用的一个极为广阔的方向与场景，而截止到本项目的产生，该领域的大数据应用并不出彩，数据化方向也比较松散，用户想要查询某方面信息只能通过各方面搜集数据进行拼凑，而且最终的评价结果也不尽如人意。因此我们想，能不能构建这样一个科技资源的搜索引擎，对科技资源进行挖掘与可视化以满足用户这方面需求呢。

带着这样一个想法，通过指导老师和实验室学长的帮助，我们取得了千万数量级的论文专利等科研信息，通过对这些信息做分词索引，最终形成了现在这样从人物、机构、领域三方面进行分析的搜索引擎，由于搜索引擎的特性，会针对所输入关键词而切分数据，因此展示上可能没有展示出千万级数据的庞大性，但搜索引擎这一形式却能使用户了解到更多实在的信息与结论，而数据规模的庞大也能在搜索面的广泛中得以体现。

尽管本项目存在着很多不足，但其在科研资源数据挖掘领域有着极大的创新性，新兴的思路，强大的市场应用价值是有一定的借鉴价值的。

通过参与冯如杯的项目的研究，我们对网络爬虫、数据挖掘、大数据索引、相似性计算等多个方向有了更多更深的了解，并且，通过参加冯如杯，我们对于如何将科研成果应用于项目创新中有了更多的经验和思考。这次冯如杯对我们整个团队都是一次学习与历练。

参考文献

[1] Stumper, U. Influence of Noni deal Calibration Items on S-Parameter Uncertainties Applying the Solr Calibration Method. IEEE Transactions on Instrumentation and Mcasurenient, 2009, 58:36-40.

[2] 伍海洋.网页主题信息抽取系统设计与实现[D].哈尔滨工业大学,2012.

[3] 周德懋,李舟军.高性能网络爬虫:研究综述[M].计算机科学，2009，36(8)：26-29.

[4] Recep Gorguluarslan,Eui-Soo Kim,Seung-Kyum Choi et al.Reliability estimation of washing machine spider assembly via classification[J],The International Journal of Advanced Manufacturing Technology,2014,72(9/12); 1581-1591.

[5] 刘金红,陆余良.主题网络爬虫研究综述[J].计算机应用研究，2007，24(10)：26-29.

[6] 霍庆,刘培植.使用 Solr 为大数据库搭建搜索引擎[J].软件,2011,32(6): 11-14.

[7] G.S. Manku, A. Jain, A.D. Sarma. Detecting Near-Duplicates for Web Crawling. 2007.

[8] M. S. Charikar. Similarity estimation techniques from rounding algorithms. STOC’02. 2002.

[9] 吴敬龙. 金融大数据平台部分模块的设计与实现 [D].北京交通大学,2015.

[10] Andrea Garratt, Mike Jackson, Peter Burden, Jon Wallis. A survey of Alternative designs for a search engine Storage Structure. Information and Software Technology, Vol.43,No.11,Oct.2001:661-677.

[11] 李副铭.垂直搜索引擎的研究与设计 [D].电子科技大学,2009.

[12] Moses S. Charikar. Similarity estimation techniques from rounding algorithms. Proceeding STOC '02 Proceedings of the thiry-fourth annual ACM symposium on Theory of computing. 2002