**密级：商业秘密 保密期限：三年**



**硕士学位论文**



**题目： 基于参与式感知的数据处理及激励机**

**制的研究与实现**

**学 号： 2013111243**

**姓 名： 王东升**

**专 业： 通信与信息系统**

**导 师： 龚向阳**

**学 院： 网络技术研究院**

**2015年 1月 2日**

独创性（或创新性）声明

本人声明所呈交的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。尽我所知，除了文中特别加以标注和致谢中所罗列的内容以外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得北京邮电大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

申请学位论文与资料若有不实之处，本人承担一切相关责任。

本人签名： 日期：

关于论文使用授权的说明

学位论文作者完全了解北京邮电大学有关保留和使用学位论文的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属北京邮电大学。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磁盘，允许学位论文被查阅和借阅；学校可以公布学位论文的全部或部分内容，可以允许采用影印、缩印或其它复制手段保存、汇编学位论文。（保密的学位论文在解密后遵守此规定）

保密论文注释：本学位论文属于保密在 三 年解密后适用本授权书。

本人签名： 日期：

导师签名： 日期：

基于参与式感知的数据处理及激励机制的研究与实现

# 摘 要

近年来，在信息检索领域，随着传统检索模式的成熟，智能化、个性化检索等越来越受重视，逐渐成为了学术界及以工业界共同的发展目标。同时，云计算、大数据相关技术飞速发展，也为信息检索的进一步演化提供了坚实的基础。正是在这种背景之下，知识库技术在信息检索领域获得了越来越多的关注，已经成为了学术界与工业界共同关注的研究和应用热点。

目前，一些互联网公司已经基于知识库系统推出了初步的应用，例如谷歌的Knowledge Graph、搜狗的知立方、百度的知心等。这些应用虽然还处于起步阶段，但已经潜移默化的改变着传统的搜索模式。而在垂直检索领域，知识库更是有着广泛应用。谷歌学术、IMDB等等许多流行的垂直检索系统，都基于主题知识库提供了不同于通用检索的丰富的垂直检索功能。

正是由于主题知识库和垂直检索都关注于某一领域相关的实体，所以二者的配合使用，是十分自然也是合理的事情。然而还很少有文章全面描述如何设计、实现、应用一个面向垂直检索的主题知识库系统。因此，本文首先对垂直搜索及主题知识库中涉及的相关技术进行了研究及分析，包括信息检索、自然语言处理、数据集成、实体关联等等。这些都是构建主题知识库所需的必不可少的技术。然后利用这些技术给出了一个应用于视频搜索的主题知识库系统的详细设计，并且针对一些关键模块给出了实现。最后，结合实际，给出了主题知识库在视频搜索中的应用及效果分析。本文提供了一个可以借鉴的知识库构建及应用实例，为实际应用提供了参考。

关键词：垂直搜索 知识库系统 实体 数据抓取 信息抽取 数据集成

RESEARCH AND DESIGN OF KEY TECHNOLOGY

OF THE DOMAIN-SPECIFIC KNOWLEDGE BASE SYSTEM

FOR VERTICAL SEARCHING

# ABSTRACT

In recent years, with the maturity of the traditional retrieval mode, intelligent search and personalized search has get more and more attention in the field of information retrieval, becoming common development goals of the academia and industry. At the same time, the rapid development of the technology related to cloud computing and big data provide a solid foundation for the evolution of information retrieval. Technology about knowledge base system which has become a common research and application hot spots of academia and industry gains more and more attention under this background.

Currently, a number of Internet companies have launched some preliminary applications based on knowledge base system, such as Google's knowledge-graph, Sogou’s knowledge-cube, Baidu's zhixin and so on. Although still in its infancy, these applications have already changing the traditional search mode imperceptible. Knowledge base also has a wide application in the field of vertical retrieval. Many popular vertical search systems，such as Google Scholar, IMDB and so on, provide rich knowledge-based vertical search functions that unlike general search.

It is very natural and reasonable thing that using knowledge base make the vertical search better, because they are both focused on related entities in a particular field. However, surprisingly little has been published about the end-to-end process of designing, building, and using such a knowledge base system. Therefore, this article first research and analyze the relevant technologies about vertical search and domain-specific knowledge base, including information retrieval, natural language processing, data integration, entities association and so on. These technologies are essential to build the domain-specific knowledge base. With these techniques, this article then gives a detailed design of a knowledge base which is applied to the subject of the video search system. And this article gives implementation of some key modules for knowledge base system. Finally, this article gives the application of domain-specific knowledge base for video search in particular, and analyzes the effect of application. Our goal with this article is to provide a real-world case study, and to provide a reference for building and using knowledge base for real-world applications.

KEY WORDS: vertical search, knowledge base system，entity, data crawl, information extraction, data integration

# 目录

[摘 要 I](#_Toc437364268)

[ABSTRACT II](#_Toc437364269)

[目录 IV](#_Toc437364270)

[第一章 绪论 1](#_Toc437364271)

[1.1 研究背景与意义 1](#_Toc437364272)

[1.2 研究内容与目标 2](#_Toc437364273)

[1.3 论文结构 3](#_Toc437364274)

[第二章 参与式感知中的激励机制 5](#_Toc437364275)

[2.1 垂直搜索引擎概述 5](#_Toc437364280)

[2.1.1 原理和架构 5](#_Toc437364286)

[2.1.2 垂直搜索引擎评价指标 6](#_Toc437364287)

[2.2 自然语言处理技术概述 7](#_Toc437364289)

[2.2.1 分词及词库 7](#_Toc437364292)

[2.2.2 词性标注 8](#_Toc437364295)

[2.2.3 实体提取 8](#_Toc437364296)

[2.2.4 实体特征及关系发现 8](#_Toc437364297)

[2.3 知识库系统概述 9](#_Toc437364298)

[2.3.1 知识库系统概述 9](#_Toc437364300)

[2.3.2 现有的一些知识库介绍 10](#_Toc437364301)

[2.4 本文用到的开源软件和相关技术 11](#_Toc437364302)

[2.4.1 Stanford NLP 12](#_Toc437364305)

[2.4.2 Scrapy 12](#_Toc437364306)

[2.4.3 IKanalyzer分词器 13](#_Toc437364307)

[2.4.4 Lucene 14](#_Toc437364308)

[第三章 实验平台的需求分析 15](#_Toc437364309)

[3.1 主题知识库系统设计概述 15](#_Toc437364313)

[3.1.1 需求分析 15](#_Toc437364314)

[3.1.2 系统总体设计 16](#_Toc437364315)

[3.2 数据层组件设计 17](#_Toc437364316)

[3.2.1 数据层组件的设计 17](#_Toc437364317)

[3.2.2 数据层组件设计关键点 18](#_Toc437364318)

[3.3 平台层组件设计 20](#_Toc437364319)

[3.3.1 平台层组件的设计 20](#_Toc437364321)

[3.3.2 平台层组件设计中的关键点 21](#_Toc437364322)

[3.4 应用层组件设计 21](#_Toc437364323)

[3.5 监控及报表系统设计 22](#_Toc437364324)

[第四章 相关技术、设计与实现 24](#_Toc437364325)

[4.1 数据获取技术 24](#_Toc437364327)

[4.1.1 数据获取实现中的难点 24](#_Toc437364328)

[4.1.2 数据获取技术的实现 24](#_Toc437364329)

[4.2 实体关联 30](#_Toc437364330)

[4.2.1 实体关联要解决的问题 30](#_Toc437364331)

[4.2.2 实体关联的实现 31](#_Toc437364332)

[4.3 实体特征发现 34](#_Toc437364333)

[4.3.1 实体特征发现的难点 34](#_Toc437364334)

[4.3.2 实体特征发现模块的实现 34](#_Toc437364335)

[第五章 激励机制仿真实验对比 38](#_Toc437364336)

[5.1 主题知识库与垂直搜索的集成 38](#_Toc437364342)

[5.2 主题知识库在推荐中的应用 39](#_Toc437364343)

[5.2.1 推荐系统设计 39](#_Toc437364344)

[5.2.2 数据验证与测试分析 40](#_Toc437364345)

[5.3 主题知识库在检索排序中的应用 43](#_Toc437364346)

[5.3.1 应用方案设计 43](#_Toc437364347)

[5.3.2 数据验证与测试分析 43](#_Toc437364348)

[第六章 总结与展望 46](#_Toc437364349)

[6.1 工作总结 46](#_Toc437364351)

[6.2 工作展望 46](#_Toc437364352)

[参考文献 48](#_Toc437364353)

[致谢 50](#_Toc437364354)

[攻读学位期间发表的学术论文和科研情况 51](#_Toc437364355)

# 第一章 绪论

## 研究背景与意义

什么是参与式感知，来龙去脉。

参与式感知应用场景。

优点、缺点

研究热门方向

引出激励机制

## 研究内容与目标

激励机制是什么

为什么需要激励机制

现有激励机制的主要方向介绍

还有哪些问题没有解决，正是我们的研究方向

于现有垂直搜索系统中，提升垂直搜索引擎的用户体验。

## 论文结构

本文共分为6章：

介绍研究背景和意义、阐述研究内容和目标；第二章详细介绍现有的激励机制设计原理和解决的问题；第三章介绍实验平台，需求分析，设计激励（任务）方案；第四章是详细设计和实现；第五章仿真激励机制的效果。

第六章对本文的工作进行了总结，并对以后的工作进行了分析展望。

# 第二章 参与式感知中的激励机制



## 服务器定价激励机制

本文探索主题知识库在垂直搜索系统中的应用，正是垂直搜索系统垂直化、深入化的特性决定了主题知识库能够为垂直搜索提供更好的补充，在多个方面对垂直搜索引擎进行改进。所以本节对一般的垂直搜索引擎进行了分析和介绍。



### 原理和架构



图2-1垂直搜索引擎整体架构图

垂直搜索引擎通常整合某一类别的各种信息，为用户提供检索及检索相关扩展服务，是专注于某一垂直领域的专用搜索引擎，可以认为其是通用搜索引擎的细分。垂直搜索引擎与传统搜索引擎类似，需要有信息采集、分析处理、索引查询等必要功能。同时，垂直搜索区别于[通用搜索引擎](http://baike.baidu.com/view/3107065.htm)，致力解决通用搜索中查询结果多、深度不够、功能较为单一等问题，在特定的垂直场景下，提供了更为细致的功能和服务。针对某一特定的垂直领域及特定人群或者是某一特定需求提供了有很高价值的详细信息和相关周边服务。“专、精、深”是垂直搜索的显著特点，而且垂直搜索不同于通用搜索引擎涵盖了海量的各行各业各领域的各种信息，一般只面向某一垂直细分的领域或者行业，而且可以提供这一领域内深入的相关服务。垂直搜索引擎从该领域内的不同站点获取数据，然后对大量的数据进行结构化处理，得到需要的结构化的数据。之后，再对这些数据做关联、建立索引，进而为用户提供检索服务。搜索引擎本质上也是一套特别的软件系统。

和传统搜索引擎相似，垂直搜索引擎架构如图2-1所示,主要包含三个模块，即结构化数据获取模块、索引模块和检索模块。

结构化数据获取是垂直搜索引擎的基石，它的任务是从多家网站获取数据，然后采用各种信息提取技术，把所需要的信息定向、分字段的提取出来。由于垂直搜索引擎是覆盖某个领域的垂直搜索，所以一家网站肯定不能涵盖所有的搜索资源，而是需要从多家网站进行数据获取。那么，不同网站对同一个条目的描述信息可能会有不同，这时候就需要对提取出来的信息进行关联。信息关联是把不同网站对同一目标进行描述的信息关联到一起，并综合各家网站的信息，得到一个最完整、最正确的目标信息。进行完数据关联和清洗之后，一般来讲，用户就可以通过检索接口对信息进行检索了。但是，一般的数据库查询性能太差，根本不能满足垂直搜索引擎高并发的需求。通常的做法是，索引器将数据库中的信息，以定制的关键字字段作为索引项，建立倒排索引。用户通过检索接口对倒排索引进行查询比普通的数据库检索能支持更大并发。Lucene索引的存储方式也是采用的如今最为优秀的倒排方式，同时它还采用了段索引机制来快速实现索引的增量更新。建立索引的过程是这样的：首先，将数据库中的网页数据解析成文本形式；然后，对文本进行分析，通常调用分析器完成最主要的分词任务，以构建倒排索引；最后，建立索引，即按照倒排方式将各个Document加入到相应的Term列表里。在本文中，主题知识库的提出就是为了解决垂直搜索中的数据问题。将以网页为基础的垂直搜索抓变为以知识库加网页为基础的垂直检索，提供更好的垂直化检索服务。

### 垂直搜索引擎评价指标

和互联网上其他产品一样，垂直搜索引擎也需要一定的标准去评判服务水平的高低。在评判的时候，除了考虑用户体验等感情因素外，还经常需要把垂直搜索引擎的表现进行抽象、量化，并提出可以在同类产品中进行横向比较的指标。当前，垂直搜索引擎主要以一下四个指标去衡量[5]：

1. 查全率：搜索引擎找到相关文档的能力，即查询结果中相关信息与互联网中全部相关信息的比率。用户的每一次查询，如果该领域内确实有资源，就应当把结果返回给用户，这才是一个完备的垂直搜索引擎。
2. 查准率：搜索引擎过滤不相关文档的能力，即查询结果中相关信息与搜索到全部结果的比率。查准率一般与排序算法有很大的相关性，一个好的排序算法应当把用户真正想要的结果排在最前面。
3. 网页更新速度：搜索引擎更新索引库的频率，这个指标直接决定了用户访问的内容是否真实有效，而且决定了用户能否获取到最新的内容。
4. 搜索耗时：用户从提交查询请求到获取查询结果持续的时间。

## 基于逆向竞拍的激励机制

不管是垂直搜索引擎系统中的索引和检索，还是主题知识库构建都依赖于分词、信息抽取等各种自然语言处理技术。可以说，结合主题知识库的垂直搜索引擎为了达到智能化、个性化的目标，自然语言处理技术的应用是必不可少的。分词、词性标注、实体抽取、关系发现等这些自然语言处理技术占据了重要的基础地位。本节主要针对这些技术进行研究介绍。



### 分词及词库

分词是自然语言处理中最基础也是最重要的一项工作，用处也十分广泛。是文本分类聚类、信息检索、信息过滤、文献自动标引、摘要自动生成等多种信息处理系统中的关键技术。本文主要研究中文相关的分词技术，常见的中文分词算法一般涉及分词方法、分词粒度、歧义消除、新词识别几个方面。

常见的中文分词方法主要有基于词典的分词、基于语法规则的分词、基于统计的分词以及组合方法等[6]。基于词典的分词方法一般按照一定策略将待分词的句子中的子字符串与词典中的多个词条进行算法判定识别，若在词典中找到某个一致的字符串，则分词成功，否则分词失败[7]。基于词典的分词方法通常包含三个要素，即词库、扫描方向、匹配策略。目前比较成熟的几种词典分词方法包括正向最大匹配法、逆向最大匹配法、双向最大匹配法、最少切分等等。在实际分词系统中，一般把基于词典的分词结果作为初步结果，之后再对结果进行优化加工。

基于语法规则的分词方法在分词过程中使用了大量的语言知识和信息作为分词依据，尤其是用这些信息来处理分词结果的歧义裁定，类似于人类的分词在阅读时的分词方法。由于用到了句法和语义信息，所以在分词任务中带入了句法分析和语义分析的任务，无形中加重了分词的实现难度。这种方法同样需要依赖词库，但是由于中文语言系统的复杂、多变性，需要很大代价才能将中文信息处理程计算机可以处理的信息形式，因此基于语法规则的中文分词系统虽然已经有一定历史，但目前还处在试验阶段，成功案例较少。

基于统计的分词方法于近些年逐渐兴起，并得到了广泛的应用。该方法的基本思想认为词是一些汉字的组合，特别是在上下文中汉字与汉字相邻共现的概率可以较好地反映这些汉字是否可以成词的可信度，越是大概率出现在一起的几个字越可能组成一个真正的词。这种方法无需词库，在新词的识别方面有着良好的表现，但需要大量的训练语料，通过学习语料来建立模型来对新的句子进行切词。通常会采用统计词频、互信息等来建立统计模型，比如最大概率分词模型、最大熵模型、N-Gram模型等，也可以考虑几种模型的结果来综合产生结果。

不同于单独使用某种方法，目前在研究和应用中通常使用以上几种方法的组合来完成分词任务。组合方法吸取了各种方法的优势，产生一个比较良好的结果。目前的组合方法一般采用多次分词的思想，初次切分基于词库，速度快且保证一定的正确率；再次切分则一般针对较为复杂的句子，采用基于统计的方法，同时配合制定的简单语法规则，这一次的分词对消歧及新词发现效果进行改进。同时，为了满足不同应用的需要，组合方法一般在多次分词中会保留不同分词粒度的结果，以供不同需要使用。



### 词性标注

词性标注(POS tagging)是给语料库中的每个词选定一个词类或者词汇类别标记的过程。从Brown在上世纪60年代首先建立语料库开始[8]，计算机自动词性标注技术发展迅速。目前已经有了很多处于研究和应用中的词性标注技术，这些技术实现机制虽然千差万别，但基本原理理论可以大致分为几类，即基于规则的词性标注方法、基于统计的词性标注方法以及将规则和统计相结合的组合方法。

基于规则的词性标注依赖于字典及由专家制定的语言规则库。标注时候先从查字典开始，如果一个词有多个词性，则根据规则库中的规则来决定该词的词性。基于统计的词性标注方法，需要语料库的支持。一般采用不同的机器学习算法，例如隐形马尔科夫链、条件随机场等配合不同的语言模型来完成标注器的训练，然后利用标注器进行词性标注。组合方法则融合了规则和统计两种方式，既利用统计减少了大量的人工工作，又依据重要的语言规则进行修订，可以用较为合理的代价得到了令人较为满意的效果。

### 实体提取

命名实体识别(Named Entity Recognition，NER)的主要任务是识别出文本中的专有名词，例如人名、地名、组织机构等，以及有意义的时间、日期及数量短语，而且需要将这些识别出的信息加以归类[9]。命名实体的识别在国外已有一定的研究历史，但针对中文的命名实体识别研究和应用起步较晚，还处于发展阶段。

与分词和词性标注类似，命名实体识别技术的发展也经过了基于规则、基于统计以及混合方法三个阶段。基于规则的方法通常依赖于语言学专家人工构造的语言规则模板和词典，一般选用的特征包括关键字、中心词、统计信息以及标点符号等等, 同时以检测模式是否和字符串相匹配作为主要手段。基于统计机器学习的方法则主要包括隐马尔可夫模型、最大熵、支持向量机、条件随机场等常用的机器学习方法。随着计算能力的提高，目前以条件随机场应用最广。

### 实体特征及关系发现

如果把主题知识库看作一个大的图形数据结构，那么实体就是这个图上的点，而实体特征的多少代表节点所包含信息量的多少，实体间的关系则是这张图上的节点之间的边。

因此，如何从大量的非结构化或半结构化文本中自动抽取出实体所具有的特征及不同实体之间的关系对知识库能否成功构建起到了至关重要的作用。而这部分也正是知识库构建之中最困难的部分。实体特征及关系的抽取既属于信息检索的一个重要分支，也是自然语言处理的一个重要分支。早期的信息抽取系统往往依赖于专家人工设计的与领域相关的抽取模式或者规则，主要是通过模式匹配的方法建立。由于这些模式或规则通常与领域密切相关，这种基于模式匹配方法建立的信息抽取系统在不同的细分领域之间的可移植性一般较差，对应用场景要求较高，在通用性方面有着非常大的缺陷。机器学习正是在这种背景之下得到了广泛的关注，机器学习通过主动或者被动的学习方式，能够针对不同领域智能的处理问题，常见的方法主要包括基于统计、基于特征向量、基于核函数，以及混合模型等几种。虽然基于机器学习的实体特征及关系抽取已经得到了较为广泛的应用，但效果和使用的便捷性仍然不能满足日益增长的需求。尤其是由于汉语句法的多样性和灵活性，以及相关机器学习的技术在国内起步较晚，目前还有很对问题需要解决。

## 其他的重要分类中的代表方法

知识库（也称知识图谱）描述现实世界中的知识体系，包含了实体、实体属性、实体之间的关系等信息。知识库可以用于查询意图解析、暗网检索、精准广告、舆情分析监控、商品搜索、社交信息挖掘等应用中，有很高的研究价值。



### 知识库系统概述

常见的知识库可以从覆盖领域、数据整合形式等多种维度进行划分[10]。按照覆盖领域，可以分为通用知识库和主题知识库。通用知识库涵盖的主题广泛，可以用于不同领域，例如谷歌的知识图谱就是一个通用知识库。而主题知识库通常只涵盖某一领域及其相关的一些知识，一般用于某一垂直领域的服务，例如谷歌学术、一淘商品库等知识库产品。

按照数据整合形式可以分为本体型知识库和以数据源组织的知识库。本体型知识库以实体为核心将知识提取为概念、实体、特征、关系等结构化数据，加以存储利用。而以数据源组织的知识库类似于传统数据库，将不同来源得到的数据进行关联分类后存储，并不对数据中的各种概念、实体等做处理，应用效果主要依赖于检索关键字的匹配及数据源的规范程度。

本文所设计的是一个本体型的主题知识库，力图涵盖某一主题领域的各种数据，包括各种概念、相关实体等等。并且将设计的主题知识库应用于某一主题领域的垂直搜索之中。2.2节中介绍的各种技术为这种类型主题知识库的构建提供了基础。

### 现有的一些知识库介绍

早期对知识图谱的研究多出现在学术界，很多文献都介绍知识库系统（如YAGO、DBLP等）的建立和研究。2012年5月，谷歌正式推出被称为知识图谱（Knowledge Graph）的新搜索功能。知识图谱随即引起了广泛的关注，国内也有了较多的研究，百度知心、搜狗知立方等产品也陆续发布。图2-2是一些现有的知识图谱系统。

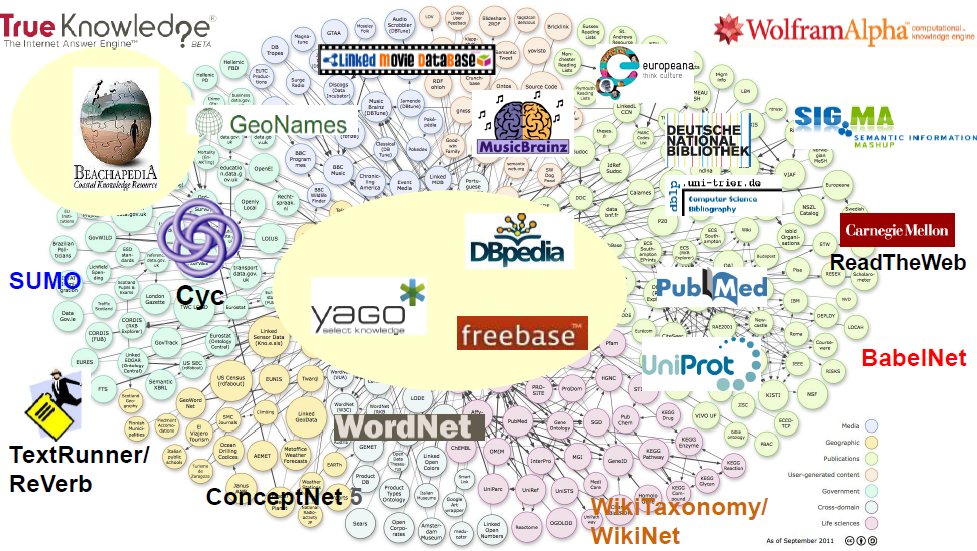


图2-2 链接的数据

2.3.2.1 YAGO系统介绍

YAGO[11]作为一个有名的知识库（知识图谱），包含了超过一千万的实体（人物、组织、城市等），以及大约一亿两千万的实体之间的关系实例。YAGO知识库是自动创建的，其数据主要有两个来源：Wikipedia和Wordnet。

YAGO用来描述实体关系的模型是基于RDFS（即资源描述框架模式）来扩展实现的。YAGO对RDFS最重要的扩展是，在RDFS之上定义了阶层化的语义数据类型以及用无向图来表示关系类型之间的转移结构。在YAGO的模型中有三种类型的实体：

1）普通实体：包括人名，地名，公司名，组织名等专有名词。

2）关系描述符：用来描述实体之间的关系类型，关系描述符同样也是一种实体。

3）关系：用来表示实体间的关系。YAGO的元关系维度可以扩展到多维，关系同样可以作为特殊的实体来参与描述更高维度的实体间关系。

YAGO从09年的第一版到现在的2s版本，发展迅速。主要有以下一些特点：1）超过95%的数据准确性，而且每一种实体关系都有对应的置信度；2）类型划分细致，根据Wikipedia和Wordnet划分了超过35万种实体类型；3）数据包含了时间和空间的属性。

2.3.2.2 DBpedia系统简介

DBpedia[12]是一个较为典型的语义网应用范例，其目标是构建一个可供人们查询的系统，系统提供从维基百科(Wikipedia)的词条里撷取出结构化的资料。DBpedia强化了针对维基百科的搜索，并将互联网上的其他资料集连结至维基百科。DBpedia基于这种理念，利用语意化技术的介入，围绕着维基百科的庞杂资讯发展了许多创新而有趣的应用，例如手机wiki版本、多面向搜寻引擎、整合wiki的地图、文件的分类与标注、词条关系查询等等。DBpedia 同时也是世界上最大的多领域知识本体库之一，而且以自由授权的方式发布，允许任何单位或者个人直接利用或者加以改进。

DBpedia目前已经发布2014版，在此版本中包含的资料集拥有超过458 万的物件，包括约144 万5000 人、73 万5000 个地点、12 万3000 张唱片、8 万7 千部电影、1 万9000 种电脑游戏、24 万1000 个组织、25 万种物种和6000 个疾病。这些资料被很多组织广泛使用，同时也是Google、百度等搜索引擎检索收录的对象。

2.3.2.3 Google 知识图谱系统简介

Google 知识图谱（Google Knowledge Graph）[13]是Google 的一个知识库，知识图谱从多种来源收集信息，这些信息被组织成知识库的形式以支持语义检索，其目标是提高Google 搜索的质量。Google 搜索在2012年开始引入了知识图谱，2012 年5 月16 日正式发布。区别于常见的搜索，集成了知识图谱的谷歌搜索除了在结果页显示搜索到的相关网站的链接列表，还在侧方提供了结构化的详细的关于检索词主题的信息。旨在使用此功能提供的信息帮助用户解决他们查询的问题，而不必跳转到结果页中其他网站来查看并汇总信息。

Google 知识图谱已经在Google 搜索中实际应用，特点鲜明。1）数据来源广泛。Google称，知识图谱的信息来自许多来源，包括CIA 的世界概况，Freebase 和维基百科等。2）知识库数据量巨大。截至2012 年，根据谷歌发布的信息，知识图谱语义网络已经包含了超过570 亿个对象，超过18 亿个介绍，而且这些不同的对象之间有数量庞大的链接关系，这些关系已经开始用来帮助谷歌搜索引擎解析搜索关键词的潜在含义。3）谷歌知识图谱依托于谷歌全球化的服务，支持如西班牙语、法语、德语、葡萄牙语、日语、俄罗斯语等等多种语言。

## 现有的总结和未来研究方向

本文的工作中，参考应用了许多优秀的开源软件，在使用的它们的时候，或者直接使用其某个功能，或者对其进行深度的定制来满足自己的需求。本节简要的对本文用到的一些开源软件和工具进行介绍。



### Stanford NLP

Stanford NLP[14]软件包是斯坦福大学自然语言小组开源的用于自然语言处理的一系列软件集。包含了分词器、词性标注器、语法分析器、命名实体识别器等常用的自然语言处理工具和部分语料库。这些工具基于统计、深度学习、规则等等实现，在研究和实际产品中有着广泛的应用。下面简单介绍本文用到的几个工具。

Stanford Part-Of-Speech Tagger，即词性标注器。这个词性标注工具采用最大熵和循环依赖网络模型实现。最大熵模型能比较有效的利用上下文信息，即是对新词也能有较好的效果。循环依赖网络模型利用了文法中的依存信息及多个词模板，针对传统的文法信息丢失做出了改善。

The Stanford Parser，即句法分析器，是基于统计实现的一个工具，采用神经网络模型实现。神经网络模型通常用网络拓扑描述网络整体结构，用节点特点来表示节点信息，用学习规则来约束网络构建。在处理速度、处理能力以及容错方面有较好的表现。

The Stanford NER，即命名实体识别器，同样基于统计原理，采用条件随机场(CRF)实现。条件随机场是一种无向图模型，近年来在自然语言处理中应用广泛，在新词发现、实体识别方面表现尤其良好。

### Scrapy



图2-3 Scrapy的整体架构图

Scrapy[15]也是一个爬虫框架，它是纯由Python编写的，相对于其他一些开源爬虫，它简单轻巧，非常方便，并且已经应用在实际生产中。Scrapy使用异步网络库Twisted来处理网络通讯，架构清晰，此外还提供了各种中间件接口，可以灵活的满足开发人员的需求。Scrapy由引擎、调度器、下载器、爬虫、管道、下载中间件、爬虫中间件、调度中间件几部分组成。Scrapy的整体架构如图2-3所示：图中数据流向由绿线表示，也是从种子链接开始，Scheduler 对下载任务进行分配，以交给 Downloader 进行下载，下载之后得到的web资源交给 Spider 进行分析，Spider 中的链接提取器组件从web资源中提取需要进一步抓取的链接，即页面中各种超链接，提取出的超链接经过滤后作为待抓取的任务被传回 Scheduler中等待执行；其他需要保存的数据，则被送到 Item Pipeline处理器那里，在Pipeline中对数据进行后期处理（分析、过滤、存储等）。另外，在数据流动的通道中还可以安装各种中间件，依照不同需求，可以进行定制化的处理。

### IKanalyzer分词器



图2-4 IKanalyzer结构图

IKanalyzer[16] 基于Java语言开发，是一个开源的、轻量级的中文分词工具包。其核心分词算法采用了正向迭代最细粒度切分算法，可以支持细粒度和智能分词两种分词模式，有较高速的处理能力。细粒度分词给出所有可能的分词结果；智能分词模式则在分词排歧义处理和数量词合并输出方面提供了基本的支持。IKanalyzer采用了多子处理器分析模式，针对英文字母、数字、中文词汇等提供不同的分词处理器进行分词处理，兼容韩文、日文字符。词典存储压缩率高，运行期占用内存较少。在词典扩展定义方面对用户十分友好，可以支持中文，英文及数字混合等多种词语。其组成部分架构如图2-4所示。

### Lucene

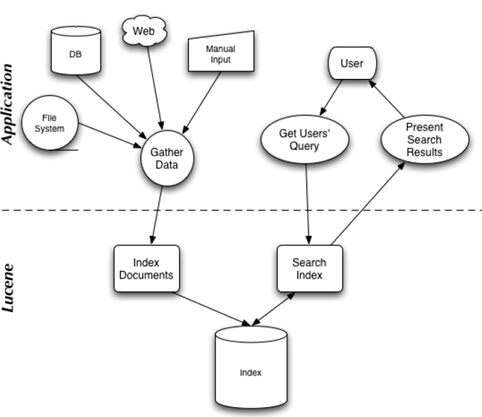


图2-5 Lucene的整体架构图

Lucene[17]是apache软件基金会的一个顶级项目，是一个开源的全文检索引擎工具，提供了完整的查询引擎和索引引擎。如图2-5示，是Lucene的整体结构，包含索引创建，索引，搜索三个核心功能。Document对象是有索引的文档在JAVA中的类型。IndexWriter负责生成索引，addDocument函数来完成从文档到索引集合的具体过程。Lucene采用了搜索中通用的倒排索引来维护索引的结构。当有搜索请求到达时，搜索语句被转换为一个Query，即查询解析器解析成的可供检索引擎理解的用户查询语句。IndexSearcher负责实现检索的过程，通过search函数来搜索Lucene的索引。IndexSearcher根据搜索得到的term weight和score算定结果集合中各条结果的排序，然后将结果返回给用户。返回给用户的排好序的文档集合用TopDocsCollector对象来表示。

# 第三章 实验平台的需求分析

垂直搜索引擎一般专注于某一领域的深度信息检索及服务，例如购物、视频、图书等领域的垂直搜索引擎。而与之对应的主题知识库，一般也是专注于相应领域的知识系统建立。结合了知识库的垂直搜索引擎不但能提供类似于通用搜索引擎的全文检索，而且还能给出结构化的，可供用户直接利用的相关信息以及基于这些信息提供的其他多种服务。



## 平台介绍

### 需求分析

随着计算机的普及，国内的互联网视频服务飞速发展，众多的视频网站有了巨大的用户数量，视频服务为互联网流量贡献了最多的数量。随着视频网站数量的增多，为了满足人们视频搜索的需求，已经出现了众多的视频垂直搜索引擎，例如搜库、360视频搜索等。笔者所在的项目组研发了一款视频垂直搜索引擎，提供了针对优酷、爱奇艺等众多视频网站视频的检索服务。

然而，不管是研究型的视频垂直搜索引擎还是市场上各家公司推出的视频搜索服务，基本都还停留在检索这一基本功能之上，而且能够检索的内容一般也局限于视频名称、演员、简单介绍等等，无法提供更高级的检索或者服务功能。除此之外，随着移动端用户数量的迅速增加，传统PC上的检索方式对移动用户的需求更加无法满足。

正是基于以上一些原因，我们希望视频搜索能够为用户提供以检索为核心的功能更加丰富的视频搜索功能，以使得视频垂直搜索具有如下功能：垂直搜索引擎系统全面、及时的采集互联网视频网站的视频信息资源，经过信息提取、存储、索引等过程，结合主题知识库提供的知识系统，为用户提供覆盖多家网站内容资源的导航服务，提供便捷、高效的信息检索服务，提供个性化、智能化、区别于一般视频搜索的一站式视频相关服务。视频领域的主题知识库建设正是为视频垂直搜索引擎更多功能的实现提供了基础支持。

为了支持垂直搜索引擎系统个性化、智能化的目标，主题知识库从功能上满足以下需求：

1. 能够提供全面、及时的视频及视频相关的各种信息，包括但不限于影视、明星、周边等等资料资讯或者信息。
2. 能够方便的整合新数据，并且以统一的格式高效存储。即便是主题知识库，其所包含的内容也十分巨大且繁杂，所以需要有一个便于扩展数据整合机制，能够高效的对不同格式、不同来源的的各种信息进行整合。
3. 能够实现对各种互联网视频信息的资源的有效关联。这里所指的信息关联的意思是对不同来源、不同格式的互联网视频信息进行结构化，然后将描述同一实体的各种资料信息关联至一起。
4. 能够针对知识库的原始数据进行加工。加工包括无效信息的过滤、数据的提取及格式的统一化。只有在统一格式的、有效的数据之上，才能进一步的扩展分析。
5. 能够将众多信息组织成以实体为中心的数据网络。这里涉及到实体提取、实体特征提取、实体关系的发现等等。这些数据是提供个性化、智能化服务的关键，也是区别于传统搜索引擎的关键之处。
6. 能够提供高效的信息查询访问接口。作为一个视频主题的知识库，不光需要提供视频或者视频相关的信息，还需要把这些信息以合适的方式以服务化的方式提供出来供应用访问。信息访问应该支持检索和定制化的接口，能够批量的导出统一格式的数据，或者查询符合要求的指定数据。

### 系统总体设计

上文描述了面向垂直搜索的主题知识库所需解决的一些需求，本小节根据需求来描述面向视频检索的主题知识库系统的总体设计。一个良好的设计能为系统的扩展、维护提供极大的方便。本系统采用分层次设计，各层只关注本层的核心功能，各层之间通过约定接口来互通。这样设计，给予每层高度的定制化空间，且系统扩展及变化都十分方便。



图3-1 主题知识库系统总体设计

如图3-1主要包括数据层、平台层、应用层以及监控及报表系统。数据层负责为知识库提供数据来源，分为数据采集和数据处理两个子层；数据采集子层负责数据抓取、数据过滤、数据标注、数据导入等功能。采集层为整个知识库系统提供了最基础的数据来源，应该支持主动抓取、批量导入等等多种的数据注入和扩展形式。数据处理子层则负责对采集子层收集到的数据进行一系列的预处理，例如结构化数据、非结构化挖掘、数据聚类分类及数据融合等，将各种异构数据结构化提取并存储，然后提供基本的结构化数据访问接口，供上层的平台直接访问使用。

平台层组件负责从数据中提取知识，将知识以实体为中心组成网络，并以合适的结构存储。在平台层，最主要的功能还是实现从资料到知识的转化，将大量的资料变为有分析价值的数据，以支持上层提供更加智能化的服务。这一阶段主要的功能围绕实体完成，包括完成实体识别的组件、提取实体特征的组件、发现实体关系的组件以及热点发现等专项任务相关的知识学习组件。此外，在平台层非常重要的一个功能是知识的统一存储及访问，为此设计了知识图谱存储引擎。该引擎的目标是提供统一的知识库存储管理服务，支持不同种类知识的高效存储，支持可扩展接口的知识查询接口。只有将引擎设计好，才能在主题知识库知识体系不断扩展的情景下提供高效的服务。

应用层提供数据接口和业务接口。数据接口对应用提供主题知识库存储引擎中存储的知识，分为基础数据接口和定制的业务接口。基础数据接口实体查询、实体特征查询、实体关系查询、查询改写、语义分析等与业务关系不大的数据访问接口，主要功能是给出知识库中知识的访问接口；定制的业务接口包括推荐、个性化导航、语义检索、问答等与业务相关的功能，这些业务接口需要针对业务特征对数据中的知识进行一定的分析利用才可以实现。通过应用层两种不同的接口，主题知识库可以实现数据服务和业务服务，能为不同需求的业务提供可靠的、完善的服务。

监控及报表系统主要提供统计、安全、可靠性监测方面的功能。该模块负责统一监控系统各层的运行状态和运行日志，能针对系统不同层次的状态进行监控，对知识库的应用状况给出历史及实时状态展示。

## 需求

数据层是整个系统的基础，为知识图谱系统的构建提供了最基本的数据。采集子层和处理子层共同为知识库提供数据保障，没有数据层，知识库就没有了知识的来源，上层的体系建设更无从谈起。

### 数据层组件的设计

数据层组件的设计如图3-2所示，采集层完成数据抓取和导入，处理层完成数据的处理。通过数据层可以将互联网或者其他来源的各种资料统一收集和处理，为知识库提供基础数据支持。该层的设计重点是数据收集的全面性和实时性，以及数据处理高效性和准确性。

图3-2 数据层组件设计

### 数据层组件设计关键点



图3-3 下载器设计

在采集子层，下载器负责从互联网抓取各种数据，如图3-3所示，下载器本质上是一个网络爬虫，需要考虑以下几个方面的问题：

* 1. URL抓取队列。抓取队列是存放待抓取或者更新的URL地址，下载器会从队列中读取URL进行页面下载抓取；
  2. 下载器调度管理。调度模块这部分针对抓取队列中的URL进行分派调度，将抓取命令分发给下载器。抓取的频率、优先级区分关系到抓取的成功率和时效性。调度器决定了抓取系统能否利充分用自身的资源，高质量的完成抓取任务。
  3. 下载器抓取模块。抓取模块负责发起http等其他请求，进行页面下载，然后或者直接将页面存储，或者发送到处理器对页面直接进行处理。下载器主要考虑死链判断、突破反抓取系统、IP资源的高效利用等。

采集子层的导入解析器则相对简单一些，这里设计为解析器插件，制定统一的输出格式接口，将不同来源不同格式的导入数据解析为统一格式输出即可。



图3-4 处理器设计

如图3-4所示，在处理子层，通过数据管道的形式来对数据实现无用信息过滤、异构数据格式化统一化的功能。处理器的核心是多种可定制的，以插件形式管理的处理管道以及管道配置管理中心。不同的处理管道实现不同的处理功能，在管道处理中可以自定义各种任务的实现，如：数据过滤管道将无效数据过滤，数据的结构化提取管道提取结构化数据，数据非结构化提取管道提取非结构化数据，数据关联管道完成数据的类别划分管理，数据融合管道进行数据对齐任务，数据格式转换管道将不同结构数据转换为同一格式等。

处理子层的管道配置管理中心则对数据进过的管道及处理顺序进行管理配置，满足不同数据的不同处理要求。在指定数据采集或者导入计划时，可以通过该模块来对数据的处理流程做出定制计划。在处理完成之后，统一结构的数据存入结构化数据存储中，以供上层访问使用。

## 平台层组件设计

平台层组件负责从数据中提取知识，将知识以实体为中心组成网络，并以合适的结构存储。数据正式在平台层完成了从资料文档到以实体为中心的只是网络的转变，可以说没有平台层就没有知识库，平台层设计是知识库构建的核心也是重点难点。



### 平台层组件的设计

如图3-5所示，平台层包括任务定义工具、任务分发引擎、处理器、存储引擎、以及提供给应用层的接口。这些组件协同配合完成针对大量资料的分析、提取、组织工作。各组件完成的工作如下：

图3-5 平台层组件设计

任务定义工具。可以通过任务定义工具定义平台层需要完成的分析处理任务。任务以规定的格式，如XML或者JSON等描述，任务定义工具负责读入并将任务解析为合适任务指令。任务指令至少应该包含任务类别、优先级、依赖关系等基本内容。解析完成后将任务指令打包发送给任务分发引擎，等待任务分发引擎调度执行。

任务分发引擎。任务分发引擎接收任务指令，完成对任务的分发和调度工作。任务的分发、调度除了需要参考任务的描述信息，即类别、优先级、依赖关系等，还需参考处理器平台的负载情况。然后将依据任务及处理器状态制定好的分配及执行计划通知处理器组中的不同处理器，由具体的处理器完成任务并给予反馈。此外任务分发引擎还应该维护任务的执行状态信息，以便于及时处理任务失败等各种异常情况。

处理器组。处理器组有一系列的处理器组成，每个处理器都可以完成特定的分析处理任务，各处理器保持统一的输入输出接口，可以定制不同的实现方式。待处理完成后通知任务分发引擎，并将处理结果直接存入存储引擎。

存储引擎。主要作用是将数据统一处理存储后，提供给上层。由于整个知识库包含的信息数据大、种类多、变化可能比较频繁，所以存储引擎应该满足效率、扩展、安全等多方面的要求。

应用接口。应用接口主要是提供平台层存储引擎中存储的各种数据，以便于应用层直接利用或者对数据加工后加以利用。

### 平台层组件设计中的关键点

平台层作为整个主题知识库系统的核心，其设计的好坏直接关系到知识库中数据的质量以及知识库使用时的效率。处理器直接关系到数据处理的质量，而存储引擎则关系到数据存储和访问的高效性、可扩展性。如何设计好处理器和存储引擎则是平台层设计的关键点。

处理器的设计中涉及到较多的自然语言处理任务，例如实体识别、实体特征识别、实体关系识别等几个关键任务。正如第二章所述，这些任务的完成一般都要用到机器学习、数据挖掘技术，对处理平台的计算能力及稳定性都有较高的要求。在笔者的工作中主要采用的Stanford NLP实验室开源的自然语言处理工具集，例如实体识别采用集成了中文分词的Stanford NER工具等。

知识库中数据在逻辑上有比较鲜明的特点，即逻辑上应该是一系列的实体，以及描述这些实体的一系列特征，还有这些实体间的关系。正因为如此，很多研究工作将知识库称之为知识图谱。基于这种特点，笔者的工作中选择将数据存储在HBase[18]数据仓库中，分两张大表保存。一张表记录实体，每行是一个实体，以及实体的特征；另一张表记录实体间的关系，每行是一对关系，并记录关系的类型。通过两张表提供查询接口给业务层使用。HBase依托于HDFS[19]有较好的扩展性、安全性，且对数据自动分片索引，能够保证在大量数据之上的查询效率。

## 应用层组件设计

应用层主要包括数据接口和业务接口。数据接口主要是讲知识库中的数据提供给消费者，核心功能是完成数据授权管理、统一查询、数据格式的统一封装、以及批量导出等功能。业务接口则与应用业务相关性较大，应用层依据具体业务需求，访问平台层得到所需的数据，利用这些数据可以处理或者改进信息检索中的各种任务。比如查询语句的解析、个性化搜索展示、搜索问题回答、推荐系统等等。不同业务应用的接口设计可能区别较大，这些接口是比数据接口更高一层的应用接口。在第六章将给出主题知识图谱系统在视频检索中的应用。

如图3-6所示，应用层包括数据授权管理、统一查询、数据格式的统一封装，以及定制的业务接口等核心组件。各组件配合完成数据的访问任务，而定制的业务接口需要根据自身业务特点利用数据接口访问得到的数据定制不同的数据处理逻辑，还会涉及到数据的筛选过滤、分析等功能。



图3-6 应用层组件设计

## 监控及报表系统设计

监控及报表系统主要提供统计、安全、可靠性监测方面的功能。监控及报表系统主要包括日志归集和实时、周期报表统计两部分子系统。

日志归集系统负责收集知识库系统所有大小系统产生的日志，并进行规整化。如图3-7所示，App指所有子系统，Agent是部署在这些子系统机器上的日志收集代理，Admin管理节点负责制定Agent日志收集行为，Collector负责对日志进行归集。收集到的日志统一格式后存入HDFS分布式文件系统之中。

而报表子系统则负责产生报表数据，供系统开发、维护和使用人员了解系统状况，也可以支持各层中需要动态调度决策的模块，例如平台层的任务分发模块，所需的系统实时状态反馈。为了满足需求，报表系统包括实时计算和离线计算两个计算模块，实时计算负责反映系统实时状态及应用实时统计，而离线计算模块负责计算统计周期报表，提供系统的历史数据统计，以支持系统改进等。



图3-7 监控及报表系统设计

监控系统的核心功能就是日志的收集和处理。日志的收集采用分层的多Master方式，在各应用上的日志收集Agent将收集到的日志按配置发送至不同的Collector，然后统一存入HDFS文件系统。整个日志收集关注的是文本的传输。

报表系统依赖于Hadoop[20]和Storm[21]。Hadoop用来做离线计算，依托于MapReduce框架充分利用集群的进行高速运算和存储，产生定期的统计数据。Storm则是一个分布式的、容错的实时计算系统，用于计算对实时性要求较高的统计结果查看，一般是各种计算量较小的任务。结果的展示采用SpringMVC搭建的Web端服务来完成展示和管理。

# 第四章 相关技术、设计与实现

主题知识库的设计实现中，涉及到较多的技术，例如如何获取相关数据、实体识别、实体特征发现、实体间关系的挖掘、面向垂直搜索的应用等。本章主要介绍实现第三章设计的主题知识库中涉及的一些关键技术，包括数据获取技术、实体关联、实体特征发现三部分的详细设计实现。



## 数据接收等api的设计

### 数据获取实现中的难点

数据获取技术是主题知识库系统数据层实现中涉及的关键技术，能否高效、全面的获取到相关的数据，直接关系的知识库的详尽、准确程度。数据抓取一般包括主动抓取和被动获得两种方式，难点通常集中在主动抓取中，主动抓取一般通过网络爬虫来完成任务。主动抓取的难点则集中在如何从不同来源得到数据这个问题上。

不同于早期各种网络爬虫模拟Http协议抓取网页即可，随着 Web2.0 技术的发展，更多的WEB开发者为了提升用户体验，吸引更多的用户，在Web设计实现中使用了越来越多的技术，数据的抓取工作遇到了越来越多的挑战。尤其在近些年，JavaScript、Ajax、jQuery、登陆验证等技术越来越广泛的使用对数据获取造成了极大的影响[22]。网络爬虫通过简单的Http请求已经不能很好的完成抓取任务，越来越多的数据需要模拟用户的访问行为才可以得到。

另一方面，数据抓取的不全面则是由于越来越严重的Deep Web[23]问题。传统的Web页面一般通过页面上超链接方式来相互连接，然而Deep Web是指那些需要通过检索等方式才能访问到的Web页面，而且这些页面一般会针对不同的检索内容、访问者呈现不同的内容，这些页面一般只是其背后一个大的数据仓库的外显，例如Wikipedia、百度百科等等站点。由于网络爬虫一般通过入口地址及超链接对网页进行抓取，Deep Web特殊的方式也对传统网络爬虫数据的抓取造成了很大影响。

### 数据获取技术的实现

为了解决主题知识库实现过程中数据获取的问题，本节设计实现了一个用于主题知识库数据获取的主题网络爬虫。正如前文介绍，爬虫的抓取目标是视频领域的各种信息，包括优酷、爱奇艺等视频网站，以及百度百科、豆瓣等等站点之上关于视频的各种信息。针对4.1.1中分析的难点，该爬虫解决各种前段技术带来的数据抓取问题，包括动态数据、突破爬取限制，以及Deep Web数据的获取问题。



图4-1 数据获取模块的实现

爬虫在Linux平台上实现，采用MySQL[24]、Redis[25]、Django[26]、Scrapy、Webkit[27]等技术共同实现。在Scrapy的框架之下，根据需求定制了各种特有组件。

如图4-1所示，系统的组成及各部分的作用如下：

* 1. Core engine是整个数据获取系统的核心，完成抓取调度的策略、数据处理控制、模板控制等多种核心功能，是系统的核心，控制了数据抓取的整个流程。
  2. 在2.4.2节中给出过Scrapy的介绍，在Scrapy轻量级、可定制的特点之上，我们集成了Webkit浏览器内核插件，使用Scrapy + Webkit的方式来完成抓取下载。Webkit重要作为Scrapy下载器的插件，当遇到动态数据时取代默认的下载器。
  3. Redis是一款开源的优秀内存数据库，支持key-value等多种形式的数据存储结构，我们采用Redis作为链接库（即抓取队列）的存储。这部分作为链接库的存储模块，包含所有的链接信息，是抓取系统的核心。
  4. MySQL数据库作为网页信息的持久化存储。采用MyBatis配合完成分表分库的存储方案。用来存储抓取到的网页原始结果，以及针对页面抽取到的特定结果。利用MySQL的Dump功能可以完成周期性的页面备份。
  5. Django则用来实现爬虫管理面板。主要功能包括配置及更新抓取模板，监控系统的任务、资源状态等。在这里使用其Admin功能，主要是用来管理一个配置数据库，控制抓取爬虫的行为。

图4-1种用红色虚线圈出的部分就是我们针对4.1.1中爬虫所需解决问题设计的模块，这些模块配合来解决动态数据、爬取限制以及Deep Web问题，下文分别介绍几个问题的具体解决。

4.1.2.1 动态数据的抓取

在项目实现的过程中，由于我们需要对大量的视频网站上的数据进行抓取，例如优酷、爱奇艺、腾讯视频等等。而这些网站为了提升用户体验，用到了较多的Javascript、Ajax等很多高级技术，相当比例的数据都是动态数据，例如用户评论、相关视频推荐链接等等。所以，简单的Http请求已经无法完整的抓取所需的数据。此外，很多站点会针对不同的用户展示很多个性化的信息，一些额外的信息只有在登录之后才能抓取得到。而模拟用户交互也是动态数据采集的一大难点。

目前比较常见的动态数据采集方式有两种。一种是针对网站动态数据请求逻辑进行分析，并且用代码模拟整个访问逻辑，然后采集访问过程中的数据。这种方式执行效率较高，但开发成本较高，需要花费大量的人力分析网站的访问逻辑，且有时需要大量试验才能完整的重现访问过程。另一种方式是采用集成Webkit等浏览器内核来模拟用户访问网站时的交互事件，并采集数据。集成浏览器内核的方式效率相对较低，但适用性比较广、且开发成本较低。

由于视频领域存在大量的站点需要抓取，针对不同站点分析访问逻辑的方法费时多、成本大、扩展性差，所以我们选取了集成Webkit内核的方式来完成抓取任务。

如图4-1中红色虚线圈出的动态数据下载模块，我们对Scrapy的源码进行修改，集成Webkit创建一个新的下载器中间件。相关代码如下：

**from** scrapy.http **import** Request, FormRequest, HtmlResponse

**import** gtk

**import** webkit

**import** jswebkit

**import** settings

**class** WebkitDownloader( object ):

**def** process\_request( self, request, spider ):

**if** spider.name **in** settings.WEBKIT\_DOWNLOADER:

**if**( type(request) **is** **not** FormRequest ):

webview = webkit.WebView()

webview.connect( 'load-finished', **lambda** v,f: gtk.main\_quit() )

webview.load\_uri( request.url )

gtk.main()

js = jswebkit.JSContext( webview.get\_main\_frame().get\_global\_context()) renderedBody = str( js.EvaluateScript( 'document.body.innerHTML' ) ) **return** HtmlResponse( request.url, body=renderedBody )

然后，在scrapy的配置项settings.py中加入如下代码，这样在整个系统启动时候，实现了Webkit下载器插件的注册，当出现动态数据时，调用下载器完成任务：

*#which spider should use WEBKIT*

WEBKIT\_DOWNLOADER=['ccb']

DOWNLOADER\_MIDDLEWARES = {

'rate\_crawler.dowloader.WebkitDownloader': 543,

}

**import** os

os.environ["DISPLAY"] = ":0"

至此，基于Scrapy的数据获取系统已经具备了动态数据的采集功能。此外，为了提供效率，动态数据的识别是需要解决的另一个问题。数据获取系统可以根据程序指定的抓取策略及抓取脚本区分动态数据和普通数据。当遇到动态数据时，调用编写好的模拟点击脚本，来完成数据的采集。

动态数据的识别筛选模块流程如图4-2所示。链接库以（URL、抓取器）的格式来存储待抓取的数据。抓取从访问链接库开始，经过一下几个流程：

1. 访问链接库，得到下一个待抓取的链接，即拿到下一条（URL、抓取器）数据。调用每个URL的对应的抓取器集合中对应的抓取器，完成URL抓取。
2. 抓取到的页面持久化保存，存入页面库。以供数据处理器使用。
3. 普通链接的提取，依据统一的正则来完成，相关规则存在配置库中。提取Http、ftp等常见的超链接。
4. 动态链接的提取，主要是js、ajax请求。依据配置库中的正则规则、请求/相应规则，自动提取动态的数据访问链接。
5. 模拟点击。依据脚本库中的配置，模拟所有符合规则的点击按钮，将请求转化为链接。
6. 链接过滤，过滤无用的广告、推广，即与抓取不想关的链接以及已经抓取过的重复链接。将其余有用链接更新加入链接库。



图4-2 动态数据筛选模块流程

4.1.2.2 Deep Web数据的抓取

构建视频主题知识库中需要大量的资料，这些资料不单来自各种视频网站，而且还包括互联网上海量的各种信息，例如百度百科中影视的介绍、豆瓣影评、用户博客评论等等。然而我们不可能对百度百科、豆瓣影评所有的页面都抓取一遍，然后再筛选出视频相关的页面，这种方式工作量太大，而且这些站点一般每有全部网页的链接地图，很多页面只能通过检索才能到达。所以采用传统的超链接遍历方式，是无法拿到这些网页的，这就是Deep Web问题。

为了解决Deep Web问题，如图4-1所示，我们在数据获取系统中设计了Deep Web处理模块，这个模块的主要功能就是利用视频领域词条库中的词条，结合Core engine给出的站点链接生成规则来直接生成网页URL，然后将URL更新至链接库等待抓取。



图4-3 Deep Web模块处理流程

图4-3是实现这一模块的流程图。大致流程如下，从视频领域词库中取出未处理的词条，然后取出站点检索链接生成规则，根据规则生成检索结果页URL，取出前N个较相关的检索结果。遍历这N个检索结果，直至结束。对每个检索结果判断其是否是视频领域相关的页面，是则存储链接库，否则丢弃。

4.1.2.3 数据获取系统类图

如图4-4，是基于Scrapy实现的本文所述的数据获取系统的类图。LinkExtractor是可以完成提取动态数据链接、模拟点击行为、生成DeepWeb链接等工作，分别由JSLinkExtractor、AjaxLinkExtractor、JSLinkExtractor、DeepWebLinkExtractor、NormalLinkExtractor等多种实现来完成；DownLoader可以完成各种类型数据的抓取，由HttpDownloader、HttpsDownloader、FtpDownloader、JsDownloader等类实现完成；Config完成配置的管理工作，包括RuleConfig、SpiderConfig等多种实现类用于规则及爬虫管理配置等等。

图4-4 数据获取系统接设计

## 激励的设计

### 实体关联要解决的问题

实体关联在主题知识库数据层处理子层和平台层的构建中都必不可少。本节设计实现的数据关联，主要解决两个问题，即实体与实体的关联和实体与资料的关联。

由于主题知识库的是以实体为和核心的组织结构，所以有统一的实体层次十分重要，现实中很多不同名字的东西其实都同指代一个实体，实体与实体的关联就是要解决这一问题，将实体按概念划分关联，而不是靠名字来区分。此外，数据采集系统一般会从多个来源得到很多资料，如何找出这些资料中一样的实体，也是实体与实体的关联问题。解决好实体的问题，是主题知识库知识体系建立的基础。

另一方面，将资料中的实体关联整理好，整个知识库知识体系由了点以后，如何将与这些点相关的资料各自找出，则是另一个需要解决的问题，即实体与资料的关联。实体的相关特征等知识信息，都需要从这些资料中得到，资料关联的准确性、全面性，直接关系到知识的准确性和丰富程度。

### 实体关联的实现

针对4.2.1种描述的问题，本节设计实现了一个用于实体关联的系统。类似于4.1节中数据获取技术的实现，实体关联系统的实现同样是面向视频领域。该系统致力于解决视频领域中实体与实体的关联和实体与资料的关联。

实体的关联不管在学术界还是产业界都是一直被研究和关注的问题。已经有分块法、排序法、索引法等一些比较常见的处理方法。基本的分块法（Basic Blocking）[28]在寻找相似记录对时，基本思路是将数据集分为多个彼此不相交的小块，通过两两比较分块内的记录来降低总体的比较次数。分块以Blocking Key作为依据，Key可以是记录的一个或多个属性值，这个值一般比较有代表性且在不同记录中区别较大，比如人名中的姓、年纪或邮政编码等。

排序法的基本思想则是将记录首先根据某一域值排序，排序法认为在排序后的序列中相似的记录会出现在临近的位置。排序法通常依据用户选定的若干个属性字符串作为键进行排序，之后采用固定区间大小的滑动窗口来找出相似的记录，在同一区间中的记录两两比较，来确定相似性。这种方法的缺点也比较明显：由于字符和单词位置对字符串排序的影响十分明显，所以相似的重复记录排在邻近的位置这一假设前提并不总是成立；另外当某一些值出现的次数很多时，即拥有相同或相似域值的记录数量超出了给定的窗口大小，会造成相似记录无法被比较，导致随后的聚类操作很难能将所有相似的重复记录完全识别出来，会造成结果的遗漏。

索引法[29]是Chaudhuri S提出一种在线的数据整合方法，是一种由代表性的数据清洗机制。索引法的处理需要有干净参照表的支持，其基本思路是基于干净的参照表数据生成一个ETI（Error Tolerance Index）索引，然后依靠这个索引来找到每一个在线输入的数据最匹配的干净记录，之后将二者的数据进行整合，从而完成对输入数据的整合和在线清洗。

针对传统几种方法的优缺点，本文提出了基于Lucene索引的分块关联法。该方法综合了分块法和索引法，分为两步关联。第一步粗粒度的筛选，利用基于Lucene的索引法找到迅速相似的实体或者资料，第二部在相似实体或者资料中用分块法确定真正关联的实体或者资料。

为了表述方便，我们将实体的资料统称为记录。当记录是实体时，它是一部视频及这个视频的描述信息，一般是我们利用4.1节中介绍的数据获取系统从视频网站上抓取的视频信息，这些视频信息一般有比较好的结构，通常包括名称、出产年份、演员、导演等比较完整的信息，可以理解为一堆词的集合；当记录是资料时，则一般是我们利用数据获取系统从百度百科、豆瓣影评及视频网站的用户评论区抓取到的文献，这些资料一般是一段一段的文字，可以理解为一堆句子的集合。在我们的设计中，将以句子形式组织的资料分词并处理后转变为与实体一样形式的记录，然后统一处理。



图4-5 数据关联模块的实现

如图4-5所示，关联需要经过转换器、索引、比较分类、审核、存入标准库几个阶段，各阶段的功能如下：

转换器。转换器是用于处理资料形式的记录的工具，目的是将以句子形式组织的资料转化为与实体形式相同的组织形式，以便于关联系统统一处理。该部分主要完成的功能是对资料进行切分，分为标题、概要、内容、评论几部分，然后对每部分的句子进行分词处理，形成以词构成的多个集合。这样，资料便转变成了与实体类似的形式。实体形式的记录本身便是以词组成的多个集合，这些集合可能包括视频名称集合、演员列表集合等等。经过转化器的处理，我们数据中每条记录都是一个集合的集合，其逻辑格式如下：

记录R：

{ 集合A：{词1、词2……}

集合B：{词1、词2……}

……

集合G：{词1、词2……}

}

Lucene索引器。关于Lucene本文研究在2.4.4中给出了基本介绍，这里不再赘述。如图4-3，在任务开始时，Lucene索引为空，每检测一条记录R，用该记录的描述信息，即多个词的集合关键域值搜索，然后将搜索结果中相关度最高的K条记录筛选出来，作为关联备选集，交由分类比较器进行准确分析。

分类比较。分类比较器在索引筛选后的小规模数据集上执行，在两个记录的描述集合之上用文本编辑距离的平均值来评判是否是一对关联记录，当高于成功阈值时匹配成功，成功匹配的记录统一计入标准信息库，而在成功阈值和失败阈值之间可能成功的记录会由人工审核确定，匹配值低于失败阈值的直接将这条记录加入索引里，这样确保了标准库的质量。

审核和存入标准库。可能匹配的记录对会交由人工审核。标准库的更新采用信息融合的方式，当信息冲突时候全部保留，确保数据的完整性，但美中不足的是可能会给标准库中带入错误信息。



图4-6 数据关联模块类图

如图4-6，是本文设计实现的数据关联模块的UML类图。LuceneIndexer封装了Lucene的索引、查询接口，提供统一的索引维护、查询接口；IndexLinker利用LuceneIndexer完成基于Lucene筛选的第一步关联；BlockMatcher用于计算两个个对应块之间的相似度；而BlockLinker利用BlockMatcher完成两个实体之间基于分块的匹配度计算；Linker中实现了两步的关联算法，DataLinker负责关联任务的调度、执行、结果统计。

采用本文设计实现的Lucene筛选加分块方式，能在效率和准确性上有较好的平衡。在筛选阶段，使用Lucene索引查找相似记录对有效地减少了记录比较的次数，时间复杂度接近于线性，同时也保证了容错率和召回率，因为是以多个域为依据查找记录，少数几个域相似程度低能通过其他相似程度高的域来弥补。而在分类比较，即分块阶段，则在校规模数据上进行准备计算，而且还加入了人工审核来弥补，对最终关联结果的准备性给出了保证。

## 技术选择（实现）

主题知识库中的实体在垂直检索中直观重要，大部分垂直检索的结果都以实体为单位来组织排列。然后，如何检索到用户真正需要的实体，同时提供关于实体更多的信息才是真正区别于传统搜索的地方。实体的特征在这些方面起到了至关重要的作用，实体特征的多少决定了一个展现给用户每个结果的详细程度，同时特征的多少还关系到实体间关系的准确性和多样性。所以实体特征发现是平台层中非常重要的一个组件。本节就描述我们在面向视频垂直检索的主题知识库中实体特征发现组件的实现。

### 实体特征发现的难点

利用本章前文介绍的数据获取技术和实体关联技术，我们已经可以将互联网上的资料划分组织成以实体及实体相关资料为单位的一个大集合。实体特征发现就是要从每一个每个实体相关资料中找到这个实体的代表性特征。

常见的特征发现一般以实体标签的方式体现，实现方法则包括人工提取及自动提取。人工提取的一般由专家根据资料或者自己的专业知识为相应实体打上数量较为有限的标签，类似于由作者选择的学术论文关键词的生成过程。这种方式生成的标签一般较为准确，但不够全面，而且成本极高，不适合大规模应用。自动提取方式则一般无需人工干预或者只需要很少的人工干预，通常综合利用规则、机器学习等多种方式进行自动化的标签提取。这种方式计算效率较高，但很多情况下准确率较差，资料数量的多少对结果的准确程度影响很大。实体特征发现的难点就是人工和自动两种方式的矛盾，如何能够协调好效率、准确性、适用范围，是能不能完成实体特征发现的关键。

### 实体特征发现模块的实现

本文采用的实现方案属于自动提取方式，基本思想是对实体相关文档中所有的词条进行重要性排序，重要性是指该词条是否能够体现改实体的特征。本文的实现方案用到了基于IKanalyzer的分词、TFIDF[30]思想、TextRank[31]、Word2Vec[32]等多种技术。

本文在2.4.3节中已经对IKanalyzer进行了简要的介绍，这里不再赘述其基本原理。如图4-7，本文实现的实体特征发现模块中分词功能采用了基于IKanalyzer的实现，实现中保留了IKanalyzer的核心分词算法，但为了满足需求，对IKanalyzer的词库管理进行了扩展修改，并且增加了词来源功能。在分词过程中，我们对字典的管理进行了类别区分，利用抓取时得到的结构化信息分为了视频名称、演员名、导演名称等。在子分词器进行分词时候，不光给出分词结果，还附带该次所属的类别。之后统一交由决策器进行歧义处理、合并结果，返回最终的结果即由词、词对应来源为单位组成。



图4-7 基于IKanalyzer的分词实现

TFIDF是一种已经被广泛应用的传统的词项加权方法，核心目标在于评价一个词对于一个文件集或一个语料库中一份文档的重要程度，即该词能否代表该文档。一般字词的重要程度随着它在文档中出现的次数成正比增加，但同时会随着它在文档集合或者语料库中总的出现的频率成反比下降。

TextRank算法是Mihalcea于2004年提出的一种无监督的标签提取方法，其基本思想类似于PageRank，首先根据词之间的共现关系构建一个词项图，认为词与词之问共现代表一种推荐关系，与重要词共现的词也重要，在这个图上经过迭代可以得到词项的重要度排序[33]。

Word2Vec是Google在2013年开源的一款将词表征为实数值向量高效工具，采用了CBOW（Continuous Bag-Of-Words，即连续的词袋模型）以及Skip-Gram两种模型。Word2vec是一个基于深度学习的训练模型，通过训练，可以把对文本内容的处理转变为在K维向量空间中的运算，通过计算向量空间上的向量的相似度来表示文本的语义特征。因此，Word2vec计算输出的空间词向量被用于多种自然语言处理相关的任务，比如聚类、找同义词性分析等。此外，Word2Vec的处理效率也非常高，Mikolov在他的论文中指出一个优化的单机版本一天可训练上千亿词。

如图4-8所示，是本文实现的实体特征发现系统，包括分词、分类、多维度特征提取、评分整合、类别重分配几个过程，最终生成实体的特征集合。各部分作用如下：

1. 分词。采用基于IKanalyzer定制的中文分词器，词典由定向从各大视频站点抓取的结构化数据生成以及通用词典生成。分词将实体相关的资料文档分为词的集合，以便于下一步处理。
2. 类别信息生成。采用Word2Vec工具对所有的文档集进行词向量分类，在实际工作中我们将此划分了大约800个类别，词结果以（词，词类别）的形式存储与文件中。在系统加载时，以key-value的形式存储于内存中以供查询。
3. 基于TFIDF及规则维度的特征生成。TFIDF维度，计算某一实体相关文档集中所有词的TFIDF权值算出所有词的排名。而规则主要起到过滤作用，基于词性、黑名单的规则过滤掉无用词。
4. 基于TextRank的特征生成。利用前文介绍的TextRank算法，计算与同一实体相关的所有词在文档集中的评分及排序。同样需要在结果中对无用词进行过滤。
5. 评分整合。本过程主要作用是用3）、4）两部分生成的特征集合融合生成特征集，在这一部设定两部分的权重，循环叠加计算，得到所有标签的综合排名及分值。
6. 类别重分配。根据2）种Word2Vec计算得到的类别信息对特征集合进行调整，基本思想是避免同类词过多而降低特征集合表示意义的丰富程度。计算规则是取前N个特征，分类后计算各部分比重，然后按比重排名重新分配各类别标签的数量。



图4-8 实体特征发现模块的实现

图4-9实体特征发现模块类图

图4-9是上文所述的实体特征发现模块的类图。FeatureDetector负责整个特征发现任务的调度、统计；TfIdfDetector采用TF-IDF的方法计算实体特征，TextRankDetector采用TextRank方法计算实体特征，Word2VecUtil封装了word2vec组件，提供词向量的分类及类别查询功能；Detector由将TfIdfDetector和TextRankDetector的计算结果进行合并操作，即merge；然后依据Filters将杂质特征过滤，最后依据Word2VecUtil提供的类别信息对结果进行调整，得到最终结果。在这里，Filters是一系列实现Filter接口的自定义过滤器。

## 测试

# 第五章 激励机制仿真实验对比

在前两章，本文介绍了主题知识库的设计及一些关键技术的实现。主题知识库的成功构建并不是结束，只有将知识库中的知识体系应用于垂直搜索之中，才能真正的体现主题知识库的价值。主题知识库在垂直检索中有着众多的应用场景，本章结合笔者实验室的一个视频垂直搜索项目，介绍了视频主题知识库在视频垂直检索中的应用。具体而言，针对推荐及检索排序两个场景做了介绍。



## 仿真场景

本节介绍视频主题知识库与视频垂直搜索引擎的集成。集成后的系统是本章所有测试的基本环境，所有的测试数据也由这个系统得出。主题知识库与垂直搜索引擎的集成主要通过开放接口的形式完成，即知识库开放知识，搜索系统利用这些知识来完善搜索中的各个环节。

图5-1 集成知识库的视频垂直检索系统

如图5-1所示，集成了主题知识库的视频垂直检索系统相对与常见的检索系统主要有以下几个不同：

1. 检索数据的来源。系统可以检索的数据来源包括两部分，一部分是互联网上海量的文档，与传统搜索引擎没有区别；另一部分是知识库应用层提供的知识体系。此外，资料的获取也依托于知识库系统的数据层来完成。
2. 知识库应用层对垂直检索引擎的支持。知识库应用层提供的接口可以帮助解析器、查询器、数据获取模块等垂直搜索的功能模块改进。对数据的来源、处理及分析提供了多种支持。
3. 扩展业务的支持。针对推荐、个性化检索等模块，知识库的应用层可以提信息补充支持，改进这些与检索相关的功能效果。

## 三种机制的介绍

### 推荐系统设计

图5-2 推荐系统模型

主题知识库对推荐系统设计的改进如图5-2所示。本节通过视频检索中的推荐系统来说明主题知识库的应用效果。视频检索的推荐系统基于协同过滤[34]实现，将实体的特征和关系加入了协同过滤中，构成了用户、视频（即实体）、视频特征（即实体特征）三种节点，用户的点击行为和视频间的关系（即实体关系）则构成了节点间的边。相对原始的协同过滤数据流图多了图书特征一种节点和图书关系一种边，丰富了数据流图。

### 数据验证与测试分析

5.2.2.1 测试数据集

本文选取了视频检索系统推荐模块中人工标注的200个推荐结果作为对比依据。每个人工标注的数据包含10部相关视频推荐。这些测试数据包含电影、电视剧、动漫、综艺四个类别，具体数量及比例如表5-1所示。在每次推荐之中，我们以视频名称为搜索词搜索，并记录结果页中的相关推荐结果，然后依据测试指标计算每个测试数据的结果好坏。另外，在转化率的评测方面，我们采用分析日志，测试数据集即是系统所有用户的点击日志。

表5-1 推荐测试数据

|  |  |  |
| --- | --- | --- |
| 类别 项目 | 数量（部） | 所占比例（%） |
| 电影 | 80 | 40 |
| 电视剧 | 70 | 35 |
| 动漫 | 30 | 15 |
| 综艺 | 20 | 10 |

5.2.2.2 测试指标

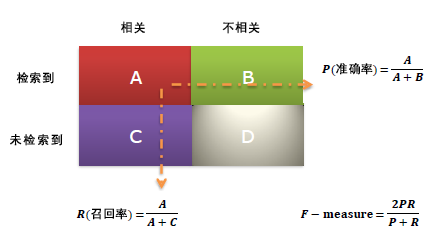


图5-2 准确率、召回率及F1值

测试指标采用通用的召回率、准确率、F1值以及转化率来评价，图5-2表示了召回率、准确率、F1值的计算规则。

召回率：指推荐结果靠前的列表中好的推荐占整个好的推荐的比例。召回率通常反映系统的全面性，即能否找到所有合理的结果。

准确率：指在位于推荐结果靠前的列表中好的推荐所占的比例。准确率通常反应系统的准确性，即能否将合理的结果排在合理的位置，即靠前的位置。

F1值：通常准确率和召回率是相互影响的，理想情况下是做到二者的均衡，即双赢。但是一般情况下，准确率高了召回率就相应的较低，而准确率低了召回率就相对较高。F1值是综合了召回率和准确率的一个评价指标，一般可以认为F1值越高，测试的结果越好。

转化率：转化率指用户点击推荐结果的比率，其评测较为特殊，我们采用系统上线后一段时间内的日志来进行评测，统计整个推荐系统中用户点击转化率的变化规律。转化率从用户的角度真实的反映了结果是否如用户期望的一致。

5.2.2.3 测试结果及分析

1. 准确率、召回率、F1值的评测

如图5-3所示，是视频搜索系统中推荐系统在利用知识库优化前后的评测对比，蓝色代表召回率、绿色代表准确率、红色代表F1值，其中较浅的颜色是未使用主题知识库的推荐，深色是使用了主题知识库之后的推荐。此外，图中包括多个类别独自的测试数据以及综合的数据。

图5-2 系统准确率、召回率及F1值对比

从图中可以看出，使用主题知识库之后，推荐系统在准确率、召回率、F1值方面的表现均好于之前。可以看出在电影及综艺两个类目，数据效果提升最为明显。对于电影，造成这种差异的原因可能是电影相关的网络资料较多，影评、剧情分析之类的数据为电影类目的实体提供了大量的视频信息之外的知识信息，这些信息能更完善的描述电影类实体，完善的信息在推荐时起到了较多的良性影响。而对于综艺节目，推荐效果的提升应该是由于综艺视频包含信息较为杂乱，涵盖内容广泛，单靠视频网站的视频描述信息并不能很好的对综艺节目进行描述，有了知识库的支持信息之后，同样扩展了综艺类实体的描述信息，在推荐时起到了作用。

此外，电视剧及动漫类目也有一定提示，但效果不如其他两个类目明显，应该是一部电视剧或者动漫包含集数较多，视频网站对每一集都已经有一定数量的描述，多集之后信息相对电影或者综艺完善程度较高。所以推荐方面提升不如另外两个类目明显，但主题知识库的引入，仍然带来了推荐效果的提升。

1. 转化率

推荐系统是区别于传统导航或者搜索的另一种浏览方式，目的是简化用户的访问代价。同时，站点通过推荐把用户尽最大可能的留在了自己站内。因此，转化率的测试就是计算运行时期用户在推荐板块的点击量与推荐信息总量的比值，该比值可以在一定程度上反映推荐结果的有效性。我们选取了系统试运行之前的10天到运行3个月后这段时间内的日志进行了分析统计，得到的推荐转化率如图5-3所示。

图5-3 系统上线后转换率曲线

从图5-3我们可以清晰的看到，系统试运行期间，转化率在不断地上升后稳定，说明系统对用户的吸引度以及用户对推荐的信任度在逐渐增强。因为在优化后的系统不止用到了知识库信息，还用到了通过用户的点击行为，所以在系统上线的前一个多月之内，随着不断增多的收集到的用户新的观看行为，系统的转换率不断提升。在将近两个月之后，转换率已经比较稳定。从图中可以看出，最终系统的转换率大概提示不到一倍，在推荐方面，主题知识库的应用效果已经十分明显。

## 结果对比

### 应用方案设计

本节通过视频垂直检索中的检索系统来说明主题知识库的应用效果。视频垂直搜索的检索系统是采用基于Lucene的Solr[35]来实现的。主题知识库通过丰富检索的索引内容和扩展改进检索的评分因子项来改善系统的搜索结果。



图5-5 检索及排序系统流程

主题知识库对检索系统设计的改进如图5-5所示，并未做过多修改，只是在索引阶段将知识库中的知识数据加入了索引。在索引中，知识库数据与视频信息标准库的数据分别存储，在检索时候权重处理有所不同。图中红色箭头代表搜索过程，虚线箭头代表索引过程，知识库数据和标准库数据两部分作为被索引的资料，是用户搜索结果的两大来源。

### 数据验证与测试分析

本文选取了视频垂直搜索系统的300次检索结果进行评测，分别针对结合了主题知识库知识的检索以及未利用主题知识库知识的检索结果进行对比评测。检索测试的300个检索结果来自从系统检索日志提取的真实的用户检索关键词记录。

类似针对检索和排序的评测，我们从准确性、多样性以及搜索的点击转化率三个维度进行评测。准确性只需满足排序前两个的结果中有正确结果即可；多样性是指检索结果中数据的区分度，结果中区别越大则多样性越好。其他标准与推荐系统测试中含义一样，可以参看5.1.2节，此处不再赘述。评测结果由人工判定，取多个人工结果的平均值作为参照标准。此外，由于视频垂直检索系统准确、丰富的特性，每个结果页显示结果较少但信息较多，我们只取前8个结果作为检索结果集来评价检索效果。

图5-6 检索及排序系统优化前后对比

如图5-6所示，未集成知识库知识的检索和集成之后的检索对比，在准确性方面并没有提示，反而有微小的下降，而多样性的评测结果则显著提升。这是由于主题知识库中带来了很多视频信息标准库中没有的资料，当用户检索时显然便可以检索到更多的内容。同时，随着检索内容的增多，准确性方面有所下降，但是由于垂直检索的特殊性，即只要实体关键信息完整，检索很难出错，所以准确性的下降并不十分明显，在接受范围之内。然后多样性效果的提示显然会给用户带来更好的体验。

如图5-7所示，是从检索系统更新前二十天到系统更新60天这段时间内的系统检索日志中分析计算，得到的用户搜索结果点击转换率的数据。可以看出，用户的搜索转换率随着知识库数据的加入，迅速提升到了97%以上。可见，主题知识库对于垂直领域的信息丰富程度提升有极大的帮助。

此外，还应注意的一点是，与推荐系统中应用效果缓慢提示然后稳定的情况不同，主题知识库在检索系统中的应用能够迅速的对系统产生影响，这是由于推荐系统中除主题知识库数据外还用到了用户点击数据，系统需要积累较多的用户点击数据才能有比较稳定的推荐结果，所以结果又一定的延迟性。而检索直接利用知识库信息，无需用户反馈，效果可以迅速体现。

图5-7检索系统优化前后用户点击率

# 第六章 总结与展望



## 工作总结

本文主要对面向垂直搜索引擎的主题知识库中的关键技术进行了全面、深入的研究，在研究基础上给出了一个主题知识库设计的例子。同时，以视频垂直检索为背景，给出了主题知识库构建中几个关键技术的实现，以及主题知识库在视频垂直检索中的几个应用场景。

随着互联网各领域垂直化服务的广泛开展，主题知识库的应用越来越广泛，关于知识库的研究受到了越来越多人的关注。本文从主题知识库构建的相关技术研究开始，研究介绍了主题知识库的设计、关键技术的实现以及主题知识库在垂直检索中的应用，对主题知识库进行了全面、深入的研究介绍。本文主要工作及成果如下：

1. 本文较为全面的介绍了知识库领域的相关信息，分析了当前主题知识库研究的进展及重要性。并针对主题知识库设计实现中涉及到的相关技术进行了研究分析，包括垂直检索、自然语言处理、知识库系统等等。

2. 本文给出了一个详细的主题知识库的设计范例，并针对设计，给出了主题知识库构建中较为核心的几项技术的实现和分析，是知识库设计实现工作的一个很好参考。知识库的设计采用了分层结构的适用于垂直搜索的原型。设计原型将知识库分为数据层、平台层、应用层以及监控及报表系统四大模块，详细分析了各模块需要完成的任务及设计中的要点难点。在几个关键技术的实现中，针对数据获取方面，主要解决了动态数据、爬取限制、Deep Web等问题；在实体关联方面，给出了实体与实体关联和实体与资料关联的统一解决方案；在实体特征发现发面，提出了基于Lucene索引加分块法的二步关联法。这些技术都是实现主题知识库中必须解决的问题，直接关系到主题知识库的准确性、全面性等问题。

3. 本文给出了一个主题知识库在垂直搜索中的应用场景及效果分析，论证了主题知识库对垂直搜索引擎功能上的积极作用。在垂直搜索系统中的推荐及检索排序两个方面进行了知识库应用的研究分析，得出了很好的结论，验证了本文设计的主题知识库系统的有效性。这也从侧面印证了主题知识库系统研究的必要性。

## 工作展望

随着近些年移动互联网热潮的兴起，各种各样的垂直化服务越来越多的出现在人们的眼中。围绕“吃喝玩乐”，专注于一个领域的互联网服务越来越智能化、个性化。在这种背景之下，主题知识库的研究和应用越来越广泛，越来越受到学术界和产业界的重视。特别是云计算、大数据技术的蓬勃发展，更是使主题知识库的构建和应用成为了可能。本文设计的主题知识库在多个方面还可以进行扩展和深入优化：

1. 围绕实体的知识体系构建。随着互联网上信息爆发式的增长，越来越多的信息可以纳入主题知识库的范围。在Deep Web中，还有大量的信息需要去挖掘获取。
2. 在知识库知识体系迭代中，如何基于各种用户的反馈数据对知识库体系进行改进也是一个有待深入研究的问题。
3. 主题知识库的构建是一个复杂的、代价高的任务，如何利用云计算、大数据技术对主题知识库的构建提供有效帮助，也是十分值得研究的一个问题。

# 参考文献

1. http://baike.baidu.com/view/7491.htm
2. http://www.soku.com/
3. http://www.qunar.com/
4. http://www.etao.com/
5. 卢亮，张博文搜索引擎原理、实践与应用. 电子工业出版社，2007，55.
6. 奉国和，郑伟．国内中文自动分词技术研究综述．图书情报工作，2011，1：41-45
7. 邓宏涛. 中文自动分词系统的设计模型[J]. 计算机与数字工程, 2005, 33(4): 138-140.
8. http://baike.baidu.com/view/11592387.htm
9. 孙镇，王惠临．命名实体识别研究进展综述．现代图书情报技术，2010，(6)：42-47
10. Omkar Deshpande, Digvijay S. Lamba, Michel Tourn, et al. 2013. Building, maintaining, and using knowledge bases: a report from the trenches. In Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data (SIGMOD '13). ACM, New York, NY, USA, 1209-1220.
11. Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In Proceedings of the 16th international conference on World Wide Web (WWW '07). ACM, New York, NY, USA, 697-706.
12. http://dbpedia.org/About
13. http://www.google.com/intl/zh-CN/insidesearch/features/search/knowledge.html
14. http://nlp.stanford.edu/software/
15. http://scrapy.org/
16. https://code.google.com/p/ik-analyzer/
17. http://lucene.apache.org/
18. http://hbase.apache.org/
19. http://baike.baidu.com/view/3061630.htm
20. http://hadoop.apache.org
21. https://storm.apache.org
22. G Pant and F Menczer. Myspiders: Evolve your own intelligent Web crawlers. Autonomous Agents and Multi-Agent Systems, 5(2):221–229, 2002.
23. Yeye He, Dong Xin, Venkatesh Ganti, et al. 2013. Crawling deep web entity pages. In Proceedings of the sixth ACM international conference on Web search and data mining (WSDM '13). ACM, New York, NY, USA, 355-364.
24. https://www.mysql.com/
25. http://redis.io/
26. https://www.djangoproject.com/
27. https://www.webkit.org/
28. Jaro M A. Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida [J]. Journal of the American Statistical Association, 1989, 84(406): 414-420.
29. Chaudhuri S, Ganjam K, Ganti V, et al. Robust and efficient fuzzy match for online data cleaning[C]. Proceedings of the 2003 ACM SIGMOD international conference on Management of data. ACM, 2003: 313-324.
30. http://en.wikipedia.org/wiki/Tf–idf
31. Mihalcea R, Tarau P. TextRank: Bringing order into texts[C]. Association for Computational Linguistics, 2004.
32. Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality[C]，Advances in Neural Information Processing Systems. 2013: 3111-3119.
33. 李鹏，王斌，石志伟. Tag-TextRank：一种基于 Tag 的网页关键词抽取方法[J]. 计算机研究与发展，2012，49（11）：2344-2351
34. 项亮．推荐系统实践[M]．北京：人民邮电出版社，2012年.
35. http://lucene.apache.org/solr/

# 致谢

时光飞逝，转眼间研究生的生活就要结束了。两年半的研究生生活对我的人生具有重要的意义。在网络技术研究院宽带中心这两年多的时间里，我的学术水平和实践能力都得到了很大的提高，同时我也结识了很多非常优秀的同学，学习到了很多珍贵的东西。最重要的是这两年的生活学习中受到的导师的谆谆教诲，这对于我来说都是巨大的收获，我将深切感激。

首先，我要感谢导师程时端教授，是您给予了我一个难得的学习和深造的机会。您渊博的知识和严谨的科研精神是我耳濡目染，使我发自内心的感到敬佩。您的学术造诣、谦虚的精神将成为我人生的标杆，不断激励我进步。

感谢项目组王洪波副教授，您在日常的学习工作中给予了我极大的帮助和指导，带领我完成多个科研项目，在论文的撰写中给我提了很多宝贵建议，在就业选择上给我提出了很多真诚可贵的建议。您对工作的认真的态度是我以后工作的榜样。

感谢项目组成员在本文工作中给予的帮助，同时感谢所有在科研工作中曾给予我无私帮助和关怀的老师和同学。

感谢宽带网研究中心全体老师、同学和工作人员。我相信在大家的不断努力下，宽带网中心的发展会越来越好，能够结识大家我感到非常的荣幸，祝愿大家工作顺利，前程似锦。

最后，感谢各位专家、教授、评委老师在百忙中审阅本文，也希望能得到各位专家老师的宝贵意见和建议。

# 攻读学位期间发表的学术论文和科研情况

论文

宋涛，王洪波. 面向垂直检索的知识图谱系统研究与设计，中国科技论文在线. 2014-11-02. 论文编号201411-16.

科研项目

* + - 1. 横向研究项目：基于云计算的垂直搜索引擎系统开发
      2. 国家 863 项目：云计算中心网络大容量交换机关键技术与系统