**西安交通大学并行计算架构与模式项目**

**基于内容的多线程视频分析与检索系统 Python-based multi-thread news crawl and analysis system**

学 院（系）： 软件学院

专 业： 软件工程

学 生 姓 名： 王 洁 王亚斌

学 号： 3118311021 3118311106

指 导 教 师： 朱 利

评 阅 教 师： 朱 利

完 成 日 期： 2018.11.20

西安交通大学

Dalian University of Technology

# 摘 要

在互联网迅速发展的今天，人们获取信息的方式从以往传统的纸质报纸、杂志、期刊到广播、电视再到现在最主流的从网络获取，已经发生了巨大变化。同时，信息大爆炸也给大众带来了诸多问题，例如，监控摄像头已遍布中国大地的每个街头，昼夜不停地监视和录像。在改善社会治安的同时，产生海量视频信息，对成千上万个监控平台进行监控将耗费大量的人力、物力和时间。在海量的视频中查找我们需要的信息，无疑是大海捞针，也给视频监控带来巨大的挑战。在此背景下，论文设计实现基于内容的多线程视频分析与检索系统，用来将用户感兴趣的视频分析处理，从数据库中获得用户感兴趣的相似片段来源等信息，为用户提供更加高效、精确的视频数据信息。

论文设计并实现了基于内容的多线程视频分析与检索系统。主要是通过SSE4指令写的多线程提取程序从视频中抽取关键帧保存到数据库中，然后用SIFT算法提取这些图片的特征点，针对这些特征点使用基于OpenMP编写的Kmeans程序进行高维聚类生成词典，将所有数据库中图片针对词典多线程生成对应的单词，最后用KD-tree进行快速检索。整个项目包括OpenMP多线程技术、机器学习算法、SIFT特征提取等相关技术。深入分析了OpenMP编程语言，着重探讨了如何用OpenMP进行多线程Kmeans聚类，大大提高了生成词典所用的时间。在特征提取方面，使用的VLFeat开源机器视觉库本身已经是多线程和SSE4优化的提取速度很快，同时又用Python脚本来进行任务分配多进程并行提取特征。KD-tree的构建是最后通过Python中Numpy、SKLearn等库来实现，能够满足实时增加节点，实时检索的要求。对于图片视觉单词生成，注意到每张图片的SIFT特征点数量是不同的，因此进行了归一化操作，能够提高检索的准确度。最后采用可视化方法对结果进行展示说明，并对系统工作进行了总结。

**关键词：视频检索；Sift算法；Kmeans算法；KDTree算法；OpenMP；多线程模型**

**Python-based multi-thread news crawl and analysis system**

# Abstract

Today, with the rapid development of the Internet, people's access to information has changed dramatically from traditional paper newspapers, magazines, periodicals, radio and television to the most mainstream online access. At the same time, the information explosion has also brought many problems to the public. For example, surveillance cameras have spread all over the streets of China, monitoring and recording around the clock. While improving social security, generating massive amounts of video information, monitoring thousands of monitoring platforms will consume a lot of manpower, material resources and time. Finding the information we need in a huge amount of video is undoubtedly a needle in a haystack, and it also poses a huge challenge to video surveillance. In this context, the thesis design implements a content-based multi-threaded video analysis and retrieval system, which is used to analyze and process the video of interest to users, obtain information such as similar fragment sources of interest from the database, and provide users with more efficient, Accurate video data information.

The paper designs and implements a content-based multi-threaded video analysis and retrieval system. The multi-thread extraction program written by the SSE4 instruction is used to extract key frames from the video and save them to the database. Then, the SIFT algorithm is used to extract the feature points of these images. For these feature points, the Kmeans program based on OpenMP is used to generate a high-dimensional clustering dictionary. The images in all the databases are generated for the dictionary multi-threaded, and finally the KD-tree is used for quick retrieval. The entire project includes OpenMP multi-threading technology, machine learning algorithms, SIFT feature extraction and other related technologies. In-depth analysis of the OpenMP programming language, focusing on how to use OpenMP for multi-threaded Kmeans clustering, greatly improving the time used to generate the dictionary. In terms of feature extraction, the VLFeat open source machine vision library itself is already multi-threaded and SSE4 optimized for fast extraction, while using Python scripts for task-distribution multi-process parallel extraction features. The construction of KD-tree is finally realized by Numpy, SKLearn and other libraries in Python, which can meet the requirements of real-time adding nodes and real-time retrieval. For the picture visual word generation, it is noted that the number of SIFT feature points of each picture is different, so the normalization operation is performed, and the accuracy of the search can be improved. Finally, the results are presented by visual methods, and the system work is summarized.

**Keywords: Video Retrieval; SIFT; Kmeans algorithm; KDtree algorithm; OpenMP;**

**Multi-threaded model;**

目 录

[摘 要 I](#_Toc16734)

[Abstract II](#_Toc8990)

[1 引言 1](#_Toc5763)

[1.1 本论文研究的背景和意义 1](#_Toc658)

[1.2 主要研究内容与工作 2](#_Toc31519)

[1.3 论文的组织结构 3](#_Toc27902)

[1.4 本章小结 3](#_Toc4030)

[2 多线程视频分析与检索系统的设计 4](#_Toc23957)

[2.1 系统总体架构设计 4](#_Toc13373)

[2.2 特征提取设计 4](#_Toc2154)

[2.2.1 爬取策略设计 4](#_Toc31077)

[4.2.2 抓取字段设计 7](#_Toc15391)

[4.2.3 爬虫的多线程设计 8](#_Toc22625)

[2.3 数据处理设计 8](#_Toc9723)

[2.3.1 数据规整 8](#_Toc9982)

[2.3.2 Kmeans聚类 8](#_Toc13233)

[2.4 新闻分析设计 9](#_Toc14037)

[4.4.1 新闻热点提取 9](#_Toc26721)

[4.4.2 新闻主题提取 11](#_Toc16074)

[4.5 字典设计 14](#_Toc26262)

[2.6 本章小结 16](#_Toc27019)

[3 视频分析与检索系统的实现 17](#_Toc16212)

[3.1 数据获取模块实现 17](#_Toc30957)

[3.1.1 关键帧的获取 17](#_Toc22130)

[5.1.2 特征提取 22](#_Toc11617)

[5.2 数据处理结果 25](#_Toc28365)

[2.3 新闻分析实现 27](#_Toc22992)

[5.3.1 新闻热点提取 27](#_Toc17694)

[5.3.2 新闻主题提取 29](#_Toc4003)

[5.4 本章小结 33](#_Toc5751)

[结 论 34](#_Toc27588)

[参 考 文 献 35](#_Toc6067)

[致 谢 36](#_Toc5329)

# 引言

## 1.1 本论文研究的背景和意义

互联网快速发展，并且已经逐步渗透到社会生活的各个角落，社会生活的基本方式已经发生了巨大变化。过去人们用诸如报纸、杂志等基于物质的信息传播、交换方式已经逐步被淘汰，这类方式俨然成为新时代的弃儿，互联网逐步取代这些方式，成为生活的必需品，网络必将成为社会的万能胶，深深地将每一个人、团体连接到一起。在网络时代里，人们的生活变得前所未有的方便快捷。随着发展需要，数以万计的网站不断被建立，信息也随之呈爆炸式发展，尤其是随着Flickr、Facebook等社交网站的流行，图像、视频、音频、文本等异构数据每天都在以惊人的速度增长。

同时近年来随着多媒体技术的进一步发展，越来越多的信息以视频形式储存、传输和表现。例如Youtube是全球最大的视频分享平台，用户量高达10亿+，每天上传的UGC和PGC都是百万级别，然而这使得人们在浩如烟海的视频信息中快速容易地获得自己感兴趣的内容变得更加困难，同时视频网站也需要基于视频内容精准向用户推送感兴趣的内容。目前，监控摄像头已遍布中国大地的每个街头，昼夜不停地监视和录像。在改善社会治安的同时，产生海量视频信息，对成千上万个监控平台进行监控将耗费大量的人力、物力和时间。在海量的视频中查找我们需要的信息，无疑是大海捞针，也给视频监控带来巨大的挑战。传统的人海战术，因效率低下以及容易错过关键目标，容易使视频监控处于“监而不控”的状态。如何化解这一危机，是现代安防的热点和难点。解决这些问题视频检索是其中的关键技术。

基于文本的图像检索方法始于上世纪70年代，它利用文本标注的方式对图像中的内容进行描述，从而为每幅图像形成描述这幅图像内容的关键词，比如图像中的物体、场景等，这种方式可以是人工标注方式，也可以通过图像识别技术进行半自动标注。在进行检索时，用户可以根据自己的兴趣提供查询关键字，检索系统根据用户提供的查询关键字找出那些标注有该查询关键字对应的图片，最后将查询的结果返回给用户。这种基于文本描述的图像检索方式由于易于实现，且在标注时有人工介入，所以其查准率也相对较高。但是这种基于文本描述的方式需要人工介入标注过程，使得它只适用于小规模的图像数据，在大规模图像数据上要完成这一过程需要耗费大量的人力与财力。

基于内容的图像检索技术便逐步建立起来，并在近十多年里得到了迅速的发展。典型的基于内容的图像检索基本框架如上图1.1所示，它利用计算机对图像进行分析，建立图像特征矢量描述并存入图像特征库，当用户输入一张查询图像时，用相同的特征提取方法提取查询图像的特征得到查询向量，然后在某种相似性度量准则下计算查询向量到特征库中各个特征的相似性大小，最后按相似性大小进行排序并顺序输出对应的图片。基于内容的图像检索技术将图像内容的表达和相似性度量交给计算机进行自动的处理，克服了采用文本进行图像检索所面临的缺陷，并且充分发挥了计算机长于计算的优势，大大提高了检索的效率，从而为海量图像库的检索开启了新的大门。

论文的主要工作是基于内容的多线程视频分析与检索系统的设计和实现。本系统使用多线程技术从海量视频中快速抽取SIFT图像特征。并编写OpenMP优化过的Kmeans来对特征进行聚类获得单词向量。最终基于相似度匹配的k近邻(KD-tree)来快速检索数据库。

总的来说，视频检索需要有高性能的多线程图像处理技术、数据处理技术、海量数据存储技术、检索技术的支持，对图像处理、特征提取、数据分析等相关技术的研究可以满足用户获取特定视频内容的需求，比如高效、全面、重点突出等。

## 1.2 主要研究内容与工作

基于内容的多线程视频分析与检索系统的研究内容主要包括以下几个方面：

（1）对视频检索系统进行了介绍，包括发展和使用的相关算法及技术；

（2）设计并行预处理流程，从视频流中获取关键帧，并对图像数据进行预处理（增强，旋转，滤波，切分等）;

（3）设计并行特征提取模块，对图像数据进行高效稳定可重复的特征提取（SIFT算法）；

（4）对图像数据库建立图像特征数据库，对数据处理模块进行规划、设计，包括数据清洗、数据处理、属性规整等功能；

（5）设计并行词袋模型提取模块，抽取检索图像特征，使用基于OpenMP编写的Kmeans构建字典，生成特征向量；

（6）设计检索模块，包含相似性度量准则，排序，搜索，使用的是KD-tree算法；

## 1.3 论文的组织结构

论文整体成文结构可分为五章，具体各章内容如下：

第一章引言。本章介绍了论文的研究背景、目的和意义，概述了网络爬虫等技术的发展。并且对本论文研究内容进行了说明，简单论述了论文的组织结构。

第二章相关技术介绍。本章主要对系统各个模块进行了说明，简要论述了网络爬虫、NoSQL 技术、 特征提取技术、主题模型，为后文系统设计和实现提供了理论基础。

第三章需求分析。本章主要对需求做了分析。

第四章新闻采集与分析系统的设计。本章主要对 系统总体架构、MongoDB数据库存储、新闻抓取模块、数据处理模块、新闻分析模块进行了设计，为实现系统提供了设计基础。

第五章新闻采集与分析系统的实现。本章实现了抓取系统，包括爬虫实现、数据处理结果、新闻热点提取、新闻主题提取。对关键词抽取算法做了重点研究，采纳基于 TF-IDF 算法的特征提取技术，实现结合新闻热度的热词提取方法；利用基于语义的LDA主题模型对新闻的主题进行提取并分析出每篇新闻在主题上的分布。

最后总结了论文工作。总结了系统在设计实现过程中遇到的困难，对未来可继续进行的研究工作进行了展望。

## 1.4 本章小结

本章介绍论文的研究背景、目的和意义，并对网络爬虫等技术的发展进行了概述。同时，对本论文研究内容进行了说明，简单论述了论文的组织结构。

2 多线程视频分析与检索系统的设计

## 2.1 系统总体架构设计

系统基本环境配置为Windows操作系统，安装MinGW并且。根据系统的功能需求，设计系统的系统组织架构如图2.1所示。下文将具体介绍各模块的设计工作。

数据检

索模块

数据处

理模块

数据获

取模块

系统环境

OpenMP API

Python解释器

调用ffmpeg

原始视频

获得关键帧

特征提取

图片处理

数据规整

Kmeans聚类

计算单词向量

构建KD—Tree

图像检索

Windows操作系统

图2.1 系统组织架构图

图2.2表示的是系统流程图：

数据获取模块

可视化展示

数据处理模块

数据检索模块

元视频

图2.2 系统流程图

## 2.2 特征提取设计

### 2.2.1 爬取策略设计

系统爬虫使用深度优先爬取策略。网页链接处理流程如图4.3所示。

开始

结束

判断链接是否符合抓取目标正则表达式

判断下载队列是否为空

爬虫所获取的网页链接

从新的网页链接中取出一条链接

执行去重和排序操作

加入到待下载队列

爬虫从待下载队列取出网页链接下载数据

否

是

抓取数据

否

是

图4.3 网页链接处理流程

一开始，爬虫获取初始链接地址，由Spider中判断目标网页链接的Xpath表达式决定是否为被需要的链接，如果是，就将其加入待下载URL队列，同时对其去重、排序，然后传递给爬虫抓取，不然就认定为无用链接进行丢弃。新链接主要指向新闻具体内容页，内容页有待提取的数据字段。爬虫将内容页链接加入待下载队列进行下一步爬取。爬虫另一项重要工作是提取内容页的字段数据[12]。对于新闻内容网页，一般包括覆盖主题的正文内容（包含文字、图片、链接等）以及其他相关新闻和广告链接。爬虫主要抓取方法是：通过制定Xpath表达式和正则表达式去抓取需要的数据，包括新闻标题、新闻正文、论文来源、发布时间、转发数量、作者、领域等。XPath提取数据的方式是路径表达式，选择XML文档中的节点来获得所需数据。这些路径表达式看起来非常像用于传统计算机文件系统的路径表达式， Xpath表达式定位目标节点，获取网页文档的字段数据。位置路径由一系列位置步骤组成。XPath包含200多个内置函数。有字符串值，数值，布尔值，日期和时间比较，节点操作，序列操作等功能。XPath表达式也可用于JavaScript，Java，XML Schema，PHP，Python，C和C ++以及许多其他语言。比如对于sciencedaily网站，其与AI相关的论文研究报道网页为：<https://www.sciencedaily.com/news/computers_math/artificial_intelligence/>，

由图4.3可以看出，新闻内容网页的链接可分为三个部分，新闻内容正文链接第一部分Xpath表达式为：xpath ( '//div [ @ class= "col-xs-6col-md-3" ]/h3/a/@href' )，第二部分latest headlines的xpath表达式为：xpath ( '//\* [ @ id="featured\_shorts" ]/li/a/@href' )，第三部分More Artificial Intelligence News Xpath表达式为：xpath ( '//\* [ @ id = "headlines"]/ul/li/a/@href' )。

对于该网站每个相关领域论文研究报道的新闻而言，内容网页都划分为这三个部分，这些Xpath表达式会被写入Scrapy工程的spiders文件中，用来提取新闻内容页面。根据以上内容，本新闻爬虫系统爬取策略设定为：

（1）设定某一领域论文新闻报道为初始爬取地址；

（2）根据Xpath表达式获取新闻内容链接，提取后，将链接加入带下载队列，并等待抓取，无用链接会被丢弃；

（3）将新闻内容链接从待下载队列中拿出，下载并提取数据[13];  
（4）待下载队列为空时，停止抓取。

本系统爬取的所有有关学术论文报道的新闻网站有（以与AI相关领域为例的初始URL）：

Sciencedaily: https://www.sciencedaily.com/news/computers\_math/artificial\_intelligence/

Phys.org:

https://phys.org/search/sort/date/all/?search=%20Artificial%20intelligence

Medicalxpress:

http://medicalxpress.com/sort/date/all//

Techxplore:

https://techxplore.com/search/sort/date/all/?search=artificial%20intelligence

科学网论文: http://paper.sciencenet.cn

中国科学院: <http://www.cas.cn/syky/>

新闻内容网页抓取Xpath表达式，详细内容见表4.1。

表4.1 内容网页urlXpath表达式

|  |  |
| --- | --- |
| 抓取的网站 | 新闻内容页Xpath表达式 |
| Sciencedaily | href1 = response.xpath( '//div[@class = "col-xs-6col-md-3"]/h3/a/@href' ).extract()  href2 = response.xpath( '//\*[@id = "featured\_shorts"]/li/a/@href' ).extract()  href3 = response.xpath( '//\*[@id = "headlines"]/ul/li/a/@href' ).extract() |
| Phys.org | hrefs = response.xpath( '//h3/a/@href' ).extract() |
| Medicalxpress | hrefs = response.xpath( '//\*[@id = "container"]/section/section/section/section/div/article/div/h3/a' ).extract() |
| Techxplore | hrefs = response.xpath( '//a[@class = "title"]/@href' ).extract() |
| 科学网论文 | hrefs = response.xpath( '//td[@valign = "top"]/a/@href' ).extract() |
| 中国科学院 | hrefs = response.xpath( '//\*[@id = "D\_mainR"]/div/div[3]/ul/li/span[2]/a/@href' ).extract() |

### 4.2.2 抓取字段设计

系统初始定义字段如表4.2所示。

表4.2 Items表

|  |  |  |
| --- | --- | --- |
| 序号 | 字段名称 | 字段含义 |
| 1 | title | 新闻标题 |
| 2 | author | 论文作者 |
| 3 | content | 新闻内容 |
| 4 | source | 论文来源 |
| 5 | date | 发布时间 |
| 6 | share | 分享转发数 |
| 7 | field | 相关领域 |

分享转发数反映了新闻的热度，时间表明论文新闻的时间特征，这两个字段为后面进行热度分析提供了条件。

### 4.2.3 爬虫的多线程设计

系统使用Scrapy架构，scrapy本身就是多线程的，保证了爬虫的多线程要求，但因为python的多线程不是很健全，所以要设定request并行数量，使之增大，同时设置DOWNLOAD\_DELAY控制速度。

## 2.3 数据处理设计

论文数据处理主要包含两个功能模块：数据清洗模块、数据规整模块。

### 2.3.1 数据规整

删除空数据文件：爬取的数据不可避免的存在字段为空的现象，这是由于网络新闻数据量大，对于一些新闻，比如科学网的论文新闻，论文来源只存在于正文之中，没有固定格式，有时甚至不提及，故此网站新闻来源字段会为空，但这些新闻仍是有用的，故对于这些新闻，在来源字段标记未知。而对于哪些正文内容为空的新闻，论文选择过滤删除。

### 2.3.2 Kmeans聚类

统一内容格式：论文抓取到的新闻内容通常包含一些HTML标签，如<p></p>、<em></em>等，系统使用[python自带re模块正则表达式去除html标签](http://blog.163.com/jackylau_v/blog/static/175754040201111511753770/" \t "_blank)。

统一时间格式：论文对抓取到的新闻统一采用“年-月-日”格式，例如“2018-4-23”，抓取的网站中需要对时间数据做统一的有Sciencedaily、Techxplore、phys.org和medical。Python为用户条新闻数据增加了新对象authority，为计算热度做好准备。

## 2.4 数据处理模块设计

新闻分析模块主要通过分析处理好的新闻，得到新闻热点并进行主题分类。新闻分析流程图如图4.4所示，首先从数据库读取数据，再分别进行对应分析，最后将结果再次存入数据库。

开始

读取数据

主题分类

热点提取

存入数据库

结束

数据库

数据库

图4.4 新闻分析流程图

### 2.4.1 数据规整

特征提取是将输入数据转换为一组能很好地表示输入数据的特征的过程。 这是降维的一种特殊形式。一般的特征提取算法包括：基于统计的特征提取。例如，基于统计词频的TF-IDF方法；基于机器学习的特征提取。例如，SVM模型、决策树、朴素贝叶斯算法等；基于复杂网络的特征提取；基于语义的特征提取等。

对于一般的新闻热点提取，比如社会类新闻，因为网络关注度高，受众广泛，因而一些新闻会在短时间内拥有大批关注，可能表现在评论量与转发量上，并且由于一般新闻每天会产生很多，所以应先过滤掉关注度小的新闻。而对于报道论文的新闻，专注报道的网站数量很少，并且相关领域论文新闻的发布速度不如社会新闻类迅速，量少，阅读人数也较少，如果同一般新闻一样考虑过滤关注度小的，会丢掉大量有用的新闻，故对于本论文抓取新闻热点提取，仅考虑网站优先级，时间，转发量和TF-IDF值。

（1） 基于统计特征的TF-IDF算法

在信息检索中， TF-IDF是一种数字统计量，主要目的是计算一个词语对集合或语料库中的文档的重要程度。一般被用作信息检索、文本挖掘和用户建模的权重因子。T逆文档频率的乘积。这两个统计数据的值可以通过多种办法确定。

TF词

3

（2） 结合新闻参数信息的热词提取

一般来说，如果一则新闻发布时间距离现在越短，网站优先级越高、转发量越大，那么其热度值就越大。总结就是，新闻的热度跟网站优先级和转发量成正比，与距离当前时间的时间差成反比。在本论文中，新闻热度由公式（4.3）计算。 

 （4.3）

具体地，shares代表新闻转发量，days代表新闻发布时间距离当前的天数，priority代表新闻网站优先级。为了规避days等于0时会犯的数学错误，公式对days统一做加1处理。

综上可知，每个单词都会获得一个自己的TF-IDF值。在得到新闻热度之后，将新闻热度heat和关键词的TF-IDF值相乘，便可计算出每个词的综合热度值，如公式（4.4）所示。 

 （4.4）

其中，表示新闻j的热度，TFIDFij指的是关键词i在新闻中的TF-IDF值，表示关键词i相对新闻j的综合热度。计算得到新闻内容中每个关键词的热度值，计算求得所有新闻关键词的热度值并进行排序，来确定热点词。取得分前20名的关键词作为关注热点。

### 2.4.2 聚类获得字典

主题建模是一种无监督机器学习，它利用集群来查找数据中的潜在变量或隐藏结构。 换句话说，这是一种用大量文本查找主题的方法。主题建模对文档聚类，非结构化文图表示特征词和主题分别按一定概率组合成隐含主题、文档。

…

…

文档

特征词

隐含主题

图4.5 LDA三层拓扑结构

对于LDA模型，涉及到的数学理论有：贝叶斯理论、抽样、分布。LDA模型在层次上，结构逻辑清晰。如LDA的概率模型图4.6。

K

M

N

主题分布

词汇分布

图4.6 LDA概率模型

LDA主题模型的文本生成过程如下；

（1）通过分布算出主题－词的概率向量。

（2）由泊松分布算出特征词总数。  
（3）通过 用词、构建词典操作；接下来依据字典，将每行新闻转换为索引的形式；计算每条新闻中每个词的TF-IDF值。

## 2.5 数据检索模块设计

Scrapy提供两种设置数据存储的方式，一种是提供csv等文本格式，可通过scrapy crawl spider – onews.csv –t scv的方式设置，也可在pipelines.py中设置，例如fp.write ( item['field'].encode('utf8')+'\t' )。如果超大量的数据存储到文件，就显得不太友好，毕竟这些数据要为我所用，因此，另一种方式是将数据存储到数据库，Scrapy提供了多种数据库API，支持数据库存储。MongoDB是最流畅的数据库之一。系统抓取到的包括新闻标题、作者如图3.3所示为MongoDB分片示意图。

集合1

1TB

256GB

256GB

256GB

256GB

图4.7 MongoDB分片示意图

本系统数据库设计的表如下所示，表4.4为新闻数据表，根据设计的字段分别为每条新闻存储包括标题、内容、时间等在内的字段；表4.5为热点词数据表；表4.6为主题数据表。

表4.4 新闻数据表

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 字段名 | 类型 | 字段意义 |
| 1 | \_id | ObjectId 对象 | 新闻编号，唯一关键字 |
| 2 | title | String | 新闻标题 |
| 3 | author | String | 论文作者 |
| 4 | content | String | 新闻内容 |
| 5 | source | String | 论文来源 |
| 6 | date | String | 发布时间 |
| 7 | share | Int | 分享转发数 |
| 8 | field | String | 相关领域 |
| 9 | priority | Int | 新闻优先级 |

表4.5 热点词数据表

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 字段名 | 类型 | 字段意义 |
| 1 | \_id | ObjectId 对象 | 单词编号，唯一关键字 |
| 2 | tfidf | Double | tfidf值 |
| 3 | hotness | Double | 新闻热度值 |
| 4 | heat | Double | 综合热度值 |

表4.6 主题数据表

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 字段名 | 类型 | 字段意义 |
| 1 | \_id | ObjectId 对象 | 主题编号，唯一关键字 |
| 2 | words | List | 主题词 |

## 2.6 本章小结

本章进行了爬虫系统的设计和数据库设计，包括数据抓取模块、数据处理模块和数据分析模块的设计，为之后实现爬虫和数据处理、分析工作提供了设计基础。

3 视频分析与检索系统的实现

## 3.1 数据获取模块实现

本节将根据上节视频检索系统设计，结合程序代码详细介绍系统的实现过程。

### 3.1.1 关键帧的获取

关键帧，是指动画中一个绘图，定义任何的起点和终点平滑过渡，一系列关键帧定义了观看者将看到的运动，而关键帧在电影，视频或动画上的位置定义了运动的时间。

整个关键帧获取其实是使用FFmpeg软件来处理的，FFmpeg 是一个自由软件，可以运行音频和视频多种格式的录影、转换、流功能，同时该软件是当今各大主流播放器的内核。在编译过程通过配置，很方便地支持多线程和MMX、SSE和AVX等优化。

在本项目中通过一个C语言函数来调用FFmpeg对视频进行提取，主要目的是能自动进行任务分配并且能够同时启用多个FFmpeg进程来处理数据。关键代码如下：

for(int j=0; j<index; ++j){

**sprintf**(name[j], "ffmpeg -i %s\\%s.mp4 -r 1 %s",input\_filename, file[j], final\_out[j]);

}

#pragma omp parallel for

for(int j=0; j<index; ++j){

**system**(name[j]);

**printf**("\n");

}

### 3.1.2 关键帧的处理

获得的关键帧被按来源存放在数据库中等待预处理操作。进行特征提取之前需要对图片进行处理，主要是获得灰度图像以及尺寸变换，同时要备份一个文件列表记录需要提取的图片标识符（也就是地址），方便后面程序操作。

预处理步骤是在Python脚本convert.py中实现的，主要是使用了python自带的图像处理库PIL，因为单纯的C/C++没有能处理图片的库，这个操作本身计算量也很小，所以简单用十几行的Python代码实现比较合适，convert.py关键代码实现如下：

def **process\_image**(imagename,outimg):

*# create a pgm file*

im = Image.open(imagename).convert('L') *#.convert('L') 将RGB图像转为灰度模式*

im.save(outimg+'.pgm') *#将灰度值图像信息保存在.pgm文件中*

if \_\_name\_\_ == "\_\_main\_\_":

in\_path = '..\\CBVideoSearch\\DataSets\\Imgs\\'

out\_path = '..\\CBVideoSearch\\DataSets\\processed\_imgs\\'

for dirname in os.listdir(in\_path):

if not os.path.exists(out\_path+dirname):

os.makedirs(out\_path+dirname)

for file in os.listdir(in\_path+dirname): process\_image(in\_path+dirname+'\\'+file,str(out\_path+dirname+'\\'+file.split('.')[0]))

为方便后面处理，再转成灰度图后还使用PGM格式来保存数据，PGM是黑白超声影像图像中经常用到的格式，最大特点是文件头部分用ASCII码来存储的，方便读取。

维护的目录有三个，所有图像、已经处理过的数据、当前未处理的数据，都有TXT格式的文档，由一个Python脚本Distributer.py维护，这样后续特征提取程序可以并行地从数据库中提取数据并且处理，不会产生等待，也减少了实现进程通讯的难度。

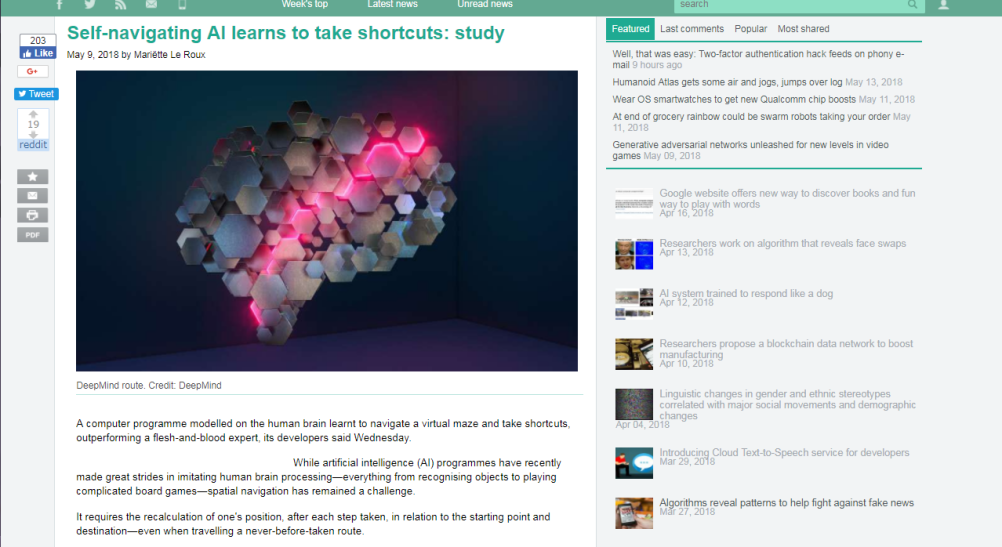


图3.2 维护的目录示意图

### 3.1.3 特征提取

本研究使用的特征是SIFT算子，SIFT（Scale-invariant feature transform）是一种检测局部特征的算法，该算法通过求一幅图中的特征点（interest points,or corner points）及其有关scale和orientation的描述子得到特征并进行图像特征点匹配，获得了良好效果。该方法的主要步骤如下：

1. 构建尺度空间
2. LoG近似DoG找到关键点
3. 给特征点赋值一个128维方向参数
4. 关键点描述子的生成



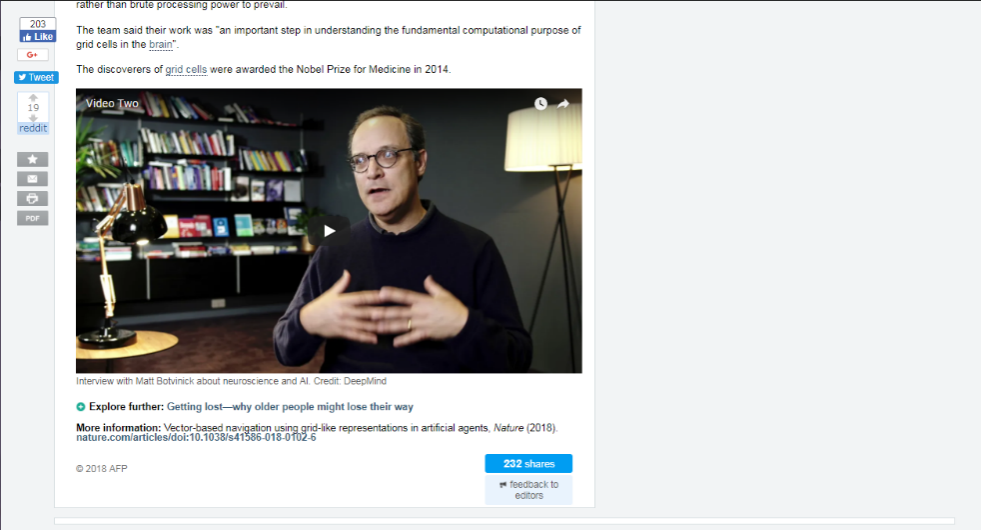


图5.5 新闻内容页

图5.6内容页要抓取的节点显示了图5.5中要抓取的内容与HTML对应关系。



图5.6 内容页要抓取的数据节点

在抓取到单条新闻链接以后，对于时间信息，需要在链接页抓取，通过调用parse\_article方法来抓取新闻的其他内容，包括标题、内容、作者、来源等内容。实现如下：

合正则表达式可完成这项任务。比如在一条新闻中，展现新闻作者、来源、发布时间信息的格式为“作者：秦庆明等来源：《环境微生物学》发布时间：2018/5/11 14:31:46”，可通过以下代码来实现抽取出作者名：

c = response.xpath ('//\*[@id = "content"]/tr[1]/td/div[1]/text()').extract()[0]

author = re.findall (r"作者：(.+?)来源",c)

在上述代码中，先使用XPath定位，提取出数据所在节点文本内容，再利用正则表达式re匹配所需信息，最终获得作者信息。经过上述实现，执行爬虫后得到数据如图5.7所示，为数据库服务器中存储的部分数据。

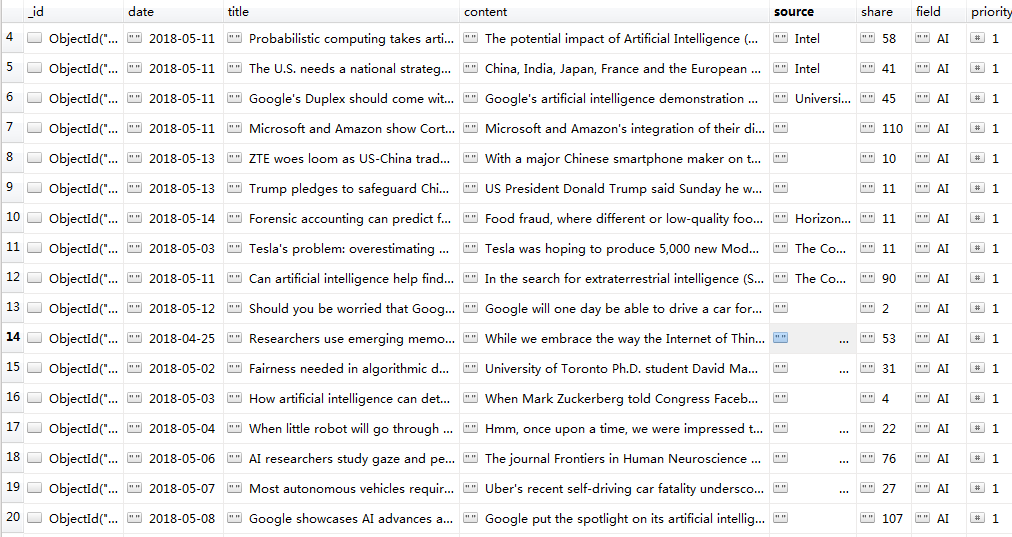


图5.7 数据库存储结果

## 3.2 Kmeans聚类生成词典

所有的数据处理都是针对数据库中的数据，所以对于每个处理都需要先连接数据库，具体实现如下：

%B%d,%Y','%Y-%m-%d')

item['date'] = time.strftime('%Y-%m-%d', time.localtime(time.time()))

经过数据处理程序后，抓取到的新闻数据拥有了统一的格式，时间格式均转换成“年-月-日”形式；新闻内容中HTML标签被清除，其余属性也拥有相同格式，具体结果实例如图5.8所示：

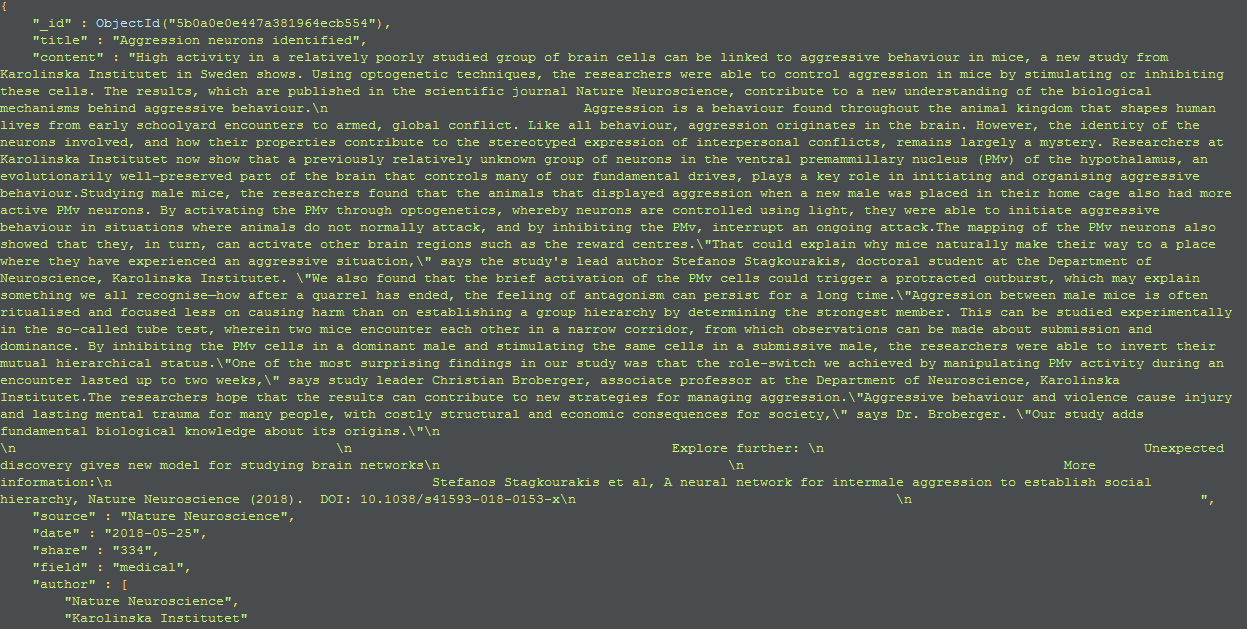


图 5.8 数据处理后的新闻详情

## 3.3 检索模块实现

### 3.3.1 KD-tree最近邻检索算法

（1）基于统计特征的TF-IDF算法

使用停用词词典、过滤标点，可以有效地提高抽提取词的有效性。如图5.9所示为关键词抽取前20名，经过去除停词、标点后能够获得真正有效关键词。

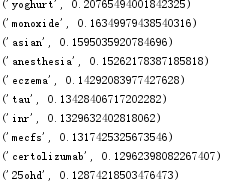
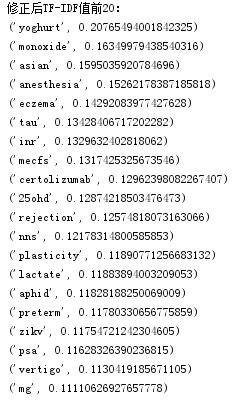
 

图5.9 关键词抽取效果

（2）结合新闻参数信息的热词提取

本文中，新闻的权重值由公式（3.1）计算，由公式（3.2）计算关键词的热度值并排序，此公式结合新闻权重和关键词自身的TF-IDF值，取排名最靠前的关键词，作为热点词。图5.10所示实例为新闻热点前20名求解结果。热度计算实现如下：

def count\_hotness():  
 for i in range(len(shares)):  
 d2 = datetime.datetime(int(times[i].split('-')[0]),int(times[i].split('-')[1]),int(times[i].split('-')[2]))  
 d = (d1-d2).days+1  
 h = (int(shares[i])/d)\*\*0.5\*int(prioritys[i])  
 hotness.append(h)

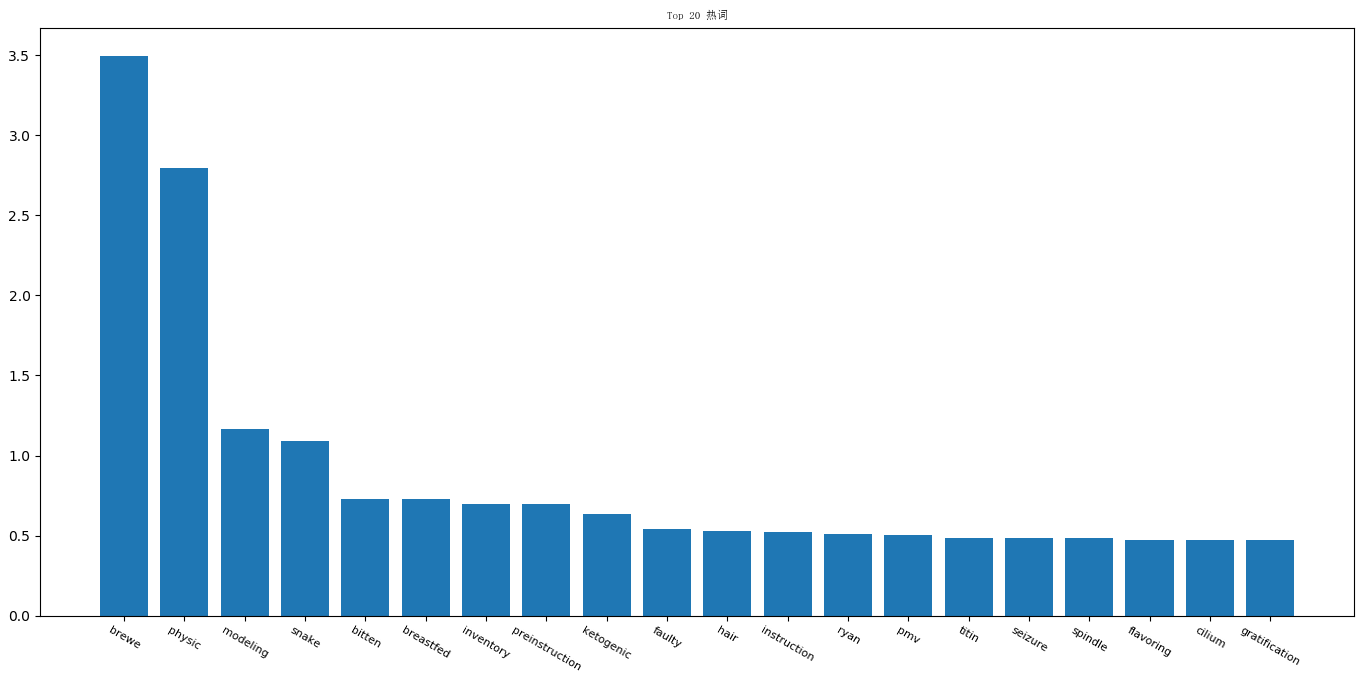


图5.10 新闻热点抽取结果图

（3）更多热点相关信息可视化实现

从图5.11可以看出关于AI的研究从1996年开始出现到现在一直成增长趋势，其中，从2011年到现在增速最快。

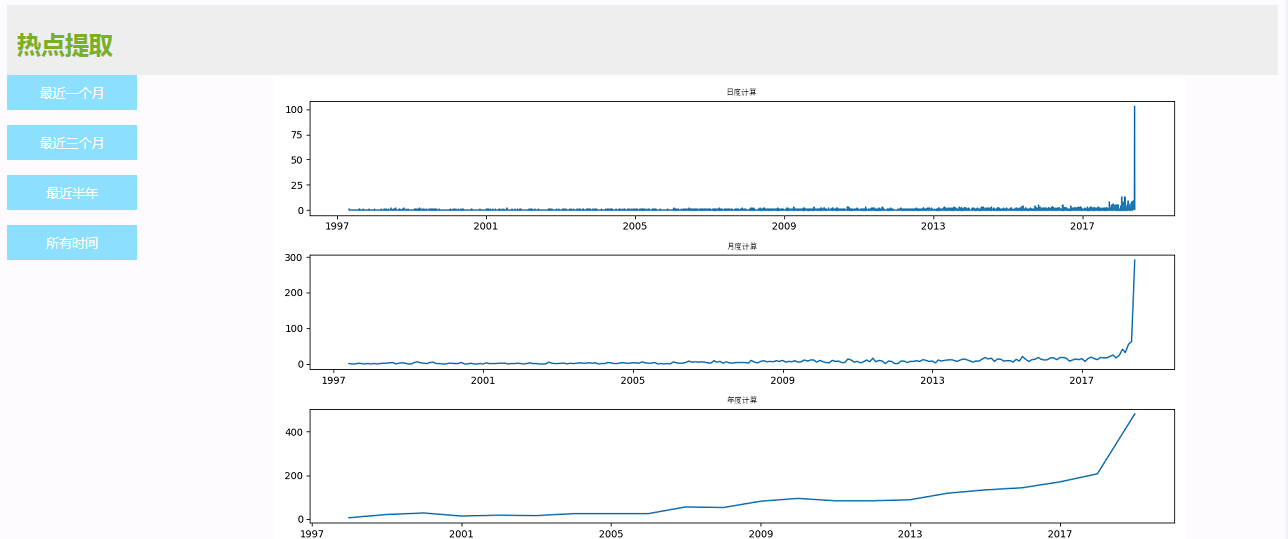


图 5.11 新闻报道折线图

### 5.3.2 新闻主题提取

LDA主题模型实现如下：

def get\_tokens(text):  
 lower = text.lower()

pprint (list(similarity))

在主题模型训练后，将主题-词语，主题-主题，主题-文档等之间的关系可视化展示，本利用pyLDAvis包以及多种可视化方式全面展示结果。pyLDAvis包探究了主题-主题，主题-词语之间的关联，主题-主题用多维标度的方式，将两者投影在低维空间，从而进行比较。主题与词语之间的关联，用了公式5.1：

（5.1）

该主题-词语关联度综合了词频和词语的独特性两种属性，其中λ是调节两种属性哪个重要的参数。另外，也展示了文档在各个主题上的概率。

主题模型训练之后，可以看见文档被分为30个主题，气泡的大小表示主题出现的概率，主题与主题之间的位置远近表达了主题之间的相近性，如图5.12左边部分所示，右侧表示的是与主题相关的词汇。

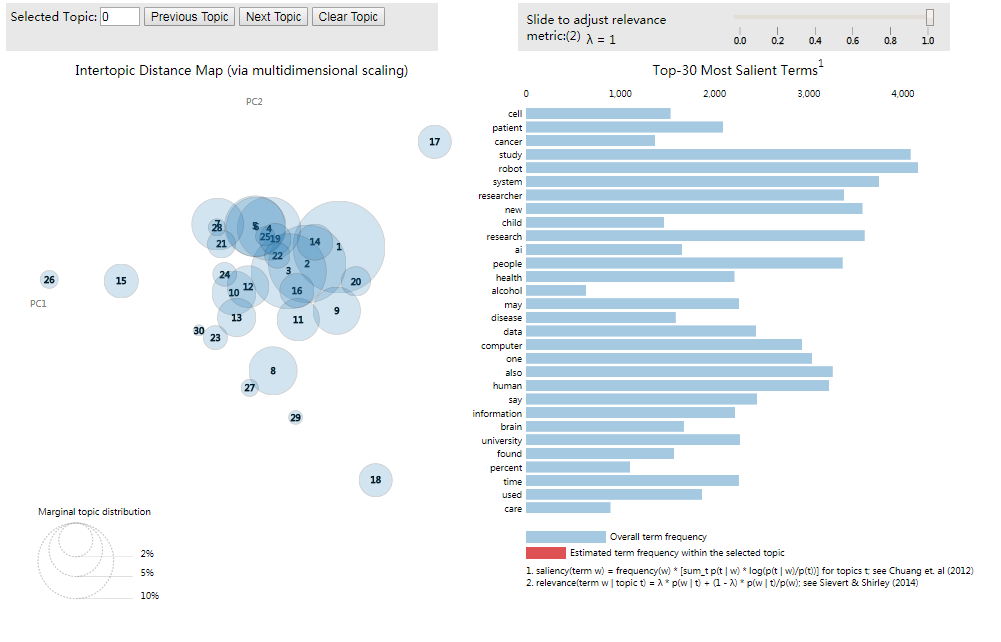


图5.12 主题与主题之间的关系

图5.13表示在某一主题下，最相关词排序情况，选取topic1为例，某个词语主题的相关性，由λ参数来调节。如果λ接近1，那么在该主题下更频繁出现的词，跟主题更相关；如果λ越接近0，那么该主题下更特殊、更独有（exclusive）的词，跟主题更相关。

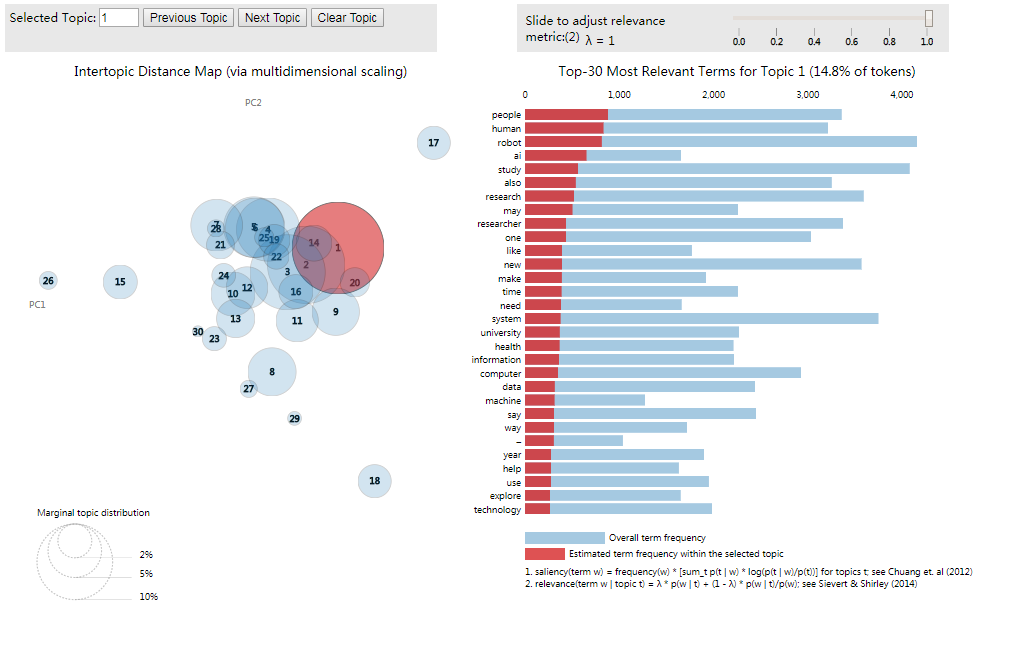


图4.13 topic 1相关词

图5.14表示某一新闻的主题的分布，选取6篇文档测试。

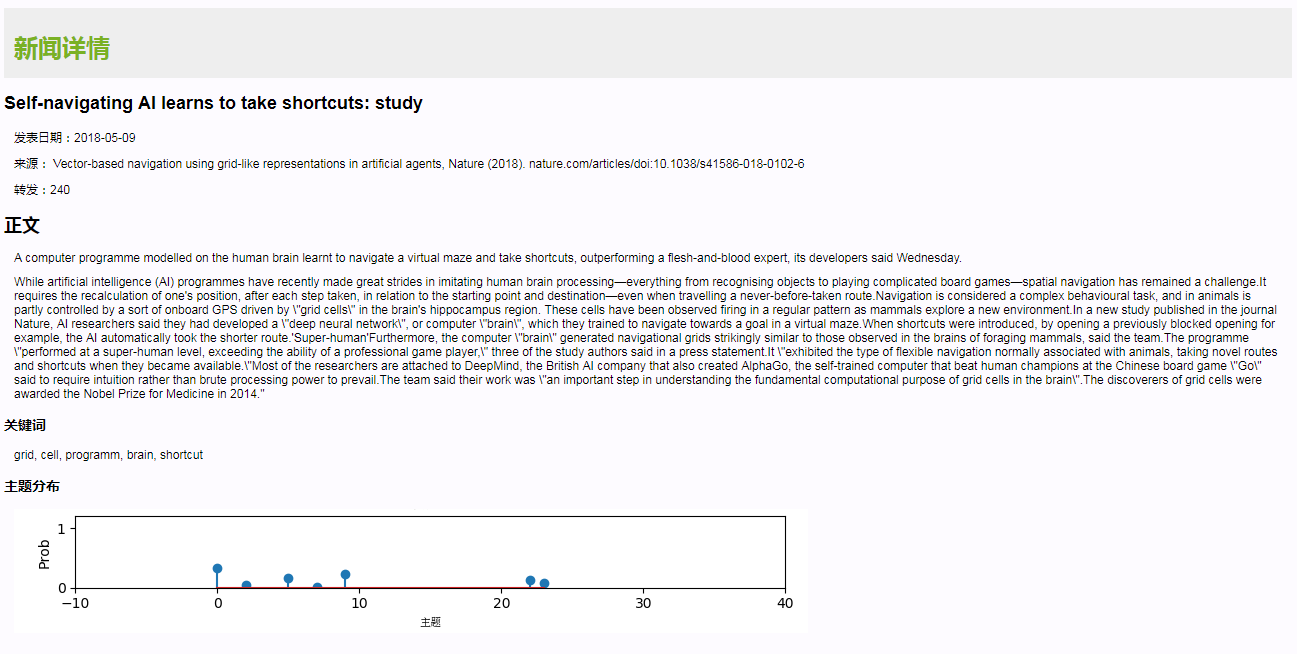


图5.14 新闻主题分布

## 3.4 本章小结

本章主要目的是为了实现新闻爬虫、新闻关注热点的提取以及新闻主题提取。对爬虫进行了实现；实现了对新闻热点的提取；利用LDA主题模型对新闻主题进行了提取。

# 结 论

论文介绍了网络爬虫的发展，阐明了其特点和原理；介绍了Scrapy 爬虫框架，包括其结构、模块功能、流程等；说明了网络新闻、新闻网页以及网络新闻爬虫的特点，并以此表明新闻抓取系统的设计总体架构，详细介绍了爬取策略、抓取字段、数据存储的设计细节；结合程序代码详细说明了爬虫编写的具体实现方法；结合实际需求，定向抓取techxplore、sciencedaily等6个新闻网站数据；设计了数据处理模块，包括数据清洗等；最后，设计数据分析模块，包括热点提取、主题提取，对于新闻热点提取，结合闻相关参数与关键词TFIDF值，实现了热点关键词的提取；再使用基于语义特征的LDA主题模型，分析出新闻主题等相关信息，并将其可视化展示出来。通过测试可以得到以下结论：

（1）使用Scrapy框架，能有效抓取到新闻的标题、正文、转发量、作者、来源、时间等内容。

（2）结合新闻相关参数和关键词TFIDF值，有效提高了新闻热点抽取准确度。

（3）采用基于语义特征的LDA主题模型，有效地将文档根据主题分类及表现主题相关性。

论文设计实现的网络新闻抓取系统、数据处理模块、数据分析模块能够达到本来目的要求，但还存在一些不足之处。首先，系统各功能模块结构松散，爬取到数据后不能够实现数据处理和数据分析的自动化；再次，新闻抓取模块和数据处理模块均使用Xpath和正则表达式来解析、抽取、识别或清洗新闻数据，这种方法有较高的效率和准确度，但当新闻站点的页面结构发生变化时，这套规则将会失效，需要重新配置。

在未来的系统研究过程中，可以根据需求完善系统的相关功能，应用机器学习等人工智能研究领域的新技术，设计实现功能全面的智能新闻抓取系统，实现数据处理和分析自动化，实现自适应的Xpath和正则表达式生成算法，使得系统更加智能，减少人工维护成本，降低新闻站点页面改版对系统运行的影响。当然论文需要改进的地方还有很多，希望可以不断改进、完善。

参 考 文 献

1. 林伟坚. 基于 Scrapy 框架的新闻实时抓取及处理系统的设计与实现[D].南开大学, 2012.
2. Baeza-Yates R, Ribeiro-Neto B. Modern information retrieval[M].//New York: ACM press, 1999.
3. 陈一峰, 赵恒凯, 余小清,等. 基于遗传算法的主题爬虫策略改进[J].计算机仿真, 2010, 第10 期:87-90.
4. 谢枫平.基于多agent强化学习的语义Web爬虫设计[J].漳州师范学院学报：自然科学版,2010,第4期:63-68.
5. 邹祥勇.传统媒体与新兴媒体融合发展的趋势与策略[J].广东广播电视大学学报,2014,23:82-85
6. Levene M. An introduction to search engines and web navigation[M].//John Wiley & Sons, 2011.
7. Abiteboul S, Preda M, Cobena G. Adaptive on-line page importance computation[C].//Proceedings of the 12th international conference on World Wide Web. ACM, 2003: 280-290.
8. Hui Wang, Jianzhuo Yan, Liying Fang. The Design and Implementation of Vertical Search Engine Based on Nutch[A]. In: Aly A. Farag, Jian Yang. Proceedings of the 3rd International Conference on Multimedia Technology (ICMT 2013)[C].// Berlin: Springer Berlin Heidelberg, 2014: 1430 - 1440.
9. LI Yue jian, ZHU Cheng rong. Study and Improvement on System Architectures of Larbin Web Crawler[J].Computer Technology & Development, 2012, 22(7): 147 - 150.
10. HB Lee, F Nazareno, SH Jung, WS Cho. A vertical search engine for school information based on Heritrix and Lucene. Lee Geuk, Howard Daniel, Dominik. Convergence and hybrid information technology[C].//Berlin: Springer Berlin Heidelberg, 2011: 344 - 351.
11. Sandra González-Bailón. Social Science in the Era of Big Data[J].Policy & Internet, 2013,5(2):147-160.
12. 崔立波.基于领域本体Web信息采集的研究[D].长春工业大学,2011.
13. 张敏,孙敏.基于Heritrix限定爬虫的设计与实现[J].计算机应用与软件,2013,30:33-35.
14. Hows D, Membrey P, Plugge E. Introduction to MongoDB[J].Mongodb Basics, 2014:1-18.

# 致 谢

在论文完成的这一刻，对我的毕业设计进行指导的老师刘宇老师、学长以和所有帮助过我的人表示最真诚的感谢。从一开始选择毕设题目、确定毕设内容到最终完成论文的整个过程，刘老师都给予了认真的指导，学长也给出了十分中肯的建议，保证了论文的顺利完成。刘老师渊博的知识、严谨的治学态度和诲人不倦的教学精神让我获益匪浅。在此，谨向刘老师表示我最衷心的谢意。

岁月如梭，大学四年转瞬即逝。我要感想我的父母，一直用满满的爱育我成才；我要感想我的大学，给予我丰富美好、人生唯有一次的大学生活；我要感想我的老师们，用“知识”的肥料孜孜不倦地灌溉我；我要感谢的同学们，陪伴我走过了四年的时光。所有美好的一切才汇聚成了现在的我，才有了现在这份毕业答卷。