文章编号：2095-2163(2021)09-0028-08 中图分类号：TP311 文献标志码：A

面向司法大数据的文本主题 OLAP 系统

王 玲[1]，刘晓清[2]，何震瀛[2]，奚军庆[3]，项 焱[4]
（1 复旦大学 软件学院，上海 200438；2 复旦大学 计算机科学技术学院，上海 200438；
3 司法部信息中心，北京 100020；4 武汉大学 法学院，武汉 430000）

摘 要：随着大数据技术的发展，加强司法大数据应用成为推进司法现代化建设的重要手段，如何处理司法大数据中的非结构化数据亟待解决。为此，本文提出了面向司法大数据的文本主题 OLAP 系统。在离线数据处理模块中，设计了 Span 数据模型，并定义了多种针对该模型的操作符；设计了基于规则的文本行政区划归类方法，并构建了主题立方体。在线上查询模块中，实现了基于倒排索引的关键词搜索方法和最大独特主题范围查询，提供了上卷、下钻、切片等功能。通过在大规模的真实数据集上对系统进行测试，实验结果证明了该系统的合理性和实用性。

关键词：大数据处理；OLAP；行政区划归类；独特主题

Big data oriented text topic OLAP system

WANG Ling[1], LIU Xiaoqing[2], HE Zhenying[2], XI Junqing[3], XIANG Yan[4]
(1 School of Software, Fudan University, Shanghai 200438, China; 2 School of Computer Science and Technology, Fudan University, Shanghai 200438, China; 3 Ministry of Justice Information Center, Beijing 100020, China; 4 School of Low, Wuhan University, Wuhan, Hubei Province 430000China)

【Abstract】 With the development of big data technology, strengthening the application of judicial big data has become an important means to promote judicial modernization. How to deal with unstructured data in judicial big data needs to be solved urgently. To this end, a text topic OLAP system oriented to judicial big data is proposed, which includes offline data processing and online query parts. In the offline data processing module, a Span data model is designed and a variety of operators for this model are defined. In addition, a rule-based text administrative division classification method is designed, and a topic cube is constructed. In the online query module, the keyword search method based on the inverted index and the largest unique subject range query are realized, and functions such as scrolling, drilling, and slicing are provided. The system is tested on a real large-scale data set, and the experimental results proves the rationality and practicability of the system.

【Key words】 big data processing; OLAP; administrative division identification; unique topic

0 引 言

随着新兴技术的发展，数据流量增长速率不断加快，大数据成为各行各业的研究热点。司法大数据的运用正成为提高司法业务效率、推进审判体系和审判能力的现代化重要手段。绝大部分的司法数据，如：法律文书、法律新闻等法律类文章都是非结构化数据，其中蕴含着案件类别、案件发生时间、案件发生地点等重要信息。如何处理这些非结构化数据，挖掘其中的重要信息亟待解决。

法律文书或法律类新闻中，往往包含地理位置信息，提取文本中的地理信息并精准定位，对案件地域分布研究具有重要意义。因此，需要设计一个能自动判别法律文章的行政区划方法，以满足用户希望根据法律文章的地点维度做数据分析的需求。

对于文本数据关键信息的获取，常见的是通过主题建模方式抽取文章的潜在主题。文本主题即文本的主旨思想，表现为一系列相关的词语。如：一篇法律文章中涉及到“离婚”这个主题，那么出现“婚姻”、“子女抚养权”、“财产纠纷”等词语的可能性会比较高。从数学角度来看，主题就是语料库中词语的条件概率分布。一个词语和主题关系越紧密，其在文章中出现的条件概率越大。主题模型是对文字中隐含主题的一种建模方法。LDA(Latent Dirichlet

基金项目：国家重点研发计划（2018YFC0830900）。

作者简介：王 玲(1996-)，女，硕士研究生，主要研究方向：数据分析；刘晓清(1964-)，男，博士，教授级高级工程师，主要研究方向：神经网络、深度学习；何震瀛(1977-)，男，博士，副教授，主要研究方向：海量数据管理、数据分析；奚军庆(1977-)，男，硕士，主要研究方向：大数据技术应用研究与分析；项 焱(1971-)，女，博士，教授，博士生导师，主要研究方向：法律援助、公益法。

通讯作者：何震瀛 Email：zhenying@ fudan.edu.cn

收稿日期：2021-06-19

Allocation)[1]是一种经典的文档主题生成模型，通过对文本进行统计分析，学习出主题的分布，可实现关键词提取及文章聚类。LDA 假设在一个文本集合中存在多个主题，每个主题下又都包含一系列的词汇。那么对集合中任意一篇文章，可看作是按照一定概率选择主题及词汇构造而成。集合中的单词构成的概率分布，组成了一个主题。不同的主题再构成一个概率分布，最终组成文章。LDA 首先按照一定概率选择某个主题，接着在此主题下以一定的概率选出某个词，作为该文章的第一个词。通过不断迭代上述步骤模拟文章生成的过程，最终得到一篇完整的文章。目前针对文章主题，各方学者展开了许多研究工作。比较常见的有主题检测与跟踪[2]、主题生命周期以及突发性[3]等。但是，这些研究都将重点放在了给定范围内文档主题的变化状态，并未考虑该主题在全局范围的地位。因此，文献[4]中提出了独特主题这一概念，用来寻找只在小范围内频繁出现而不在全部文档中出现的主题。

常见的大数据处理技术有 HBase[5]、Spark[6]等。HBase 是一个开源的非关系型分布式数据库，其参考了谷歌的 BigTable[7]建模，可以容错地存储海量稀疏的数据。Spark 是一种快速、通用、可扩展的大数据分析引擎，为分布式数据集的处理提供了一个有效的框架，并以高效的方式处理分布式数据集。Spark 实现了一种分布式的内存抽象，称为弹性分布式数据集[8]。其能够广泛适用于数据并行类应用。其中包括：批处理、结构化查询、实时流处理、机器学习与图计算等。

数据仓库技术常应用于大规模数据分析工作。OLAP (Online Analytical Processing)[9]是数据仓库[10]提供的一种对储存在数据库中的多维数据进行搜索和分析的技术，其能在不同维度上对数据进行搜索和分析。常规的 OLAP 实现方式，是用数据库中的多维数据，构建数据立方体[11]，多维数据的每一维都对应到数据立方体的每一条轴上。通过对数据立方体各个维度的组合，实现 OLAP 分析。此外，OLAP 还提供了诸如上卷、下钻、切片等操作，使用户可以更深入的分析数据。然而 OLAP 的这些操作大都只支持关系型数据。

针对上述提到的种种现象和问题，本文设计了一个面向司法大数据的文本主题 OLAP 系统。通过对文本进行预处理及分析，来实现法律文章数据行政区划归类、主题建模、独特主题查找以及基于此的 OLAP，具有重要的实际意义。本文从 3 方面对此进行了研究：

(1)针对司法数据量庞大且多以非关系型数据为主的特点，设计了一个数据模型以及基于此基础的操作符，通过各种操作符的组合达到数据处理的目的。底层使用了 Spark 分布式处理系统来实现，以此达到高效地处理大规模非关系型数据的目的。

(2)针对法律文章数据中的主题、时间、地理位置等多维度信息挖掘，设计并实现了一系列数据处理方法，使数据可以最终变为主题立方体形式，供 OLAP 查询使用。

(3)设计了基于 OLAP 的线上查询模块以获取司法数据关键信息。

1　基于 Span 数据模型的文本主题 OLAP 系统

本文设计了基于 Span 数据模型的文本主题 OLAP 系统。系统由线上查询和离线处理两部分组成，整体流程如图 1 所示。

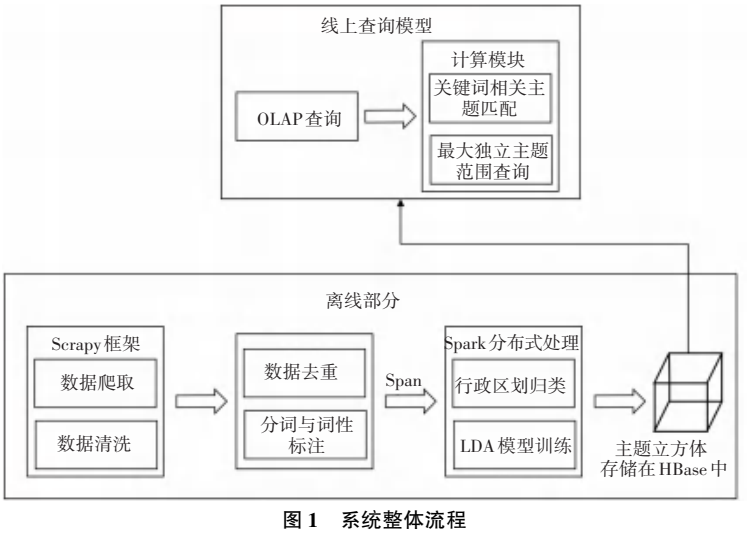

图 1　系统整体流程

Fig. 1　System flow chart

系统使用 Scrapy 分布式爬虫框架,爬取网络上的法律文章[12],利用 Spark 对文本进行清洗、去重以及分词处理[13]。本文设计了 Span 数据模型,将处理后的文本以 Span 形式存入 HBase 中。随后,使用 Span 的操作符进行关键词提取、行政区划归类以及主 LDA 模型训练,最终得到具有时间维度和地理维度的主题模型立方体。当用户发起 OLAP 查询时,系统根据用户的查询操作,结合之前保存在 HBase 中的数据以及主题立方体,进行关键词相关主题匹配和最大独立主题范围查询。用户可以对返回的结果执行上卷、下钻等 OLAP 操作来获取感兴趣的信息。

1.1 Span 数据模型

在传统的数据模型基础上,本文提出了 Span 关系模型,来描述原本非关系型的文本数据,并在此基础上设计了适用于此关系模型的操作符和任务。

系统将每篇法律文章表示为关系模型的一行,其中包含的属性见表 1。如:id 为文章的编号、title 为文章的标题、text 为文章的正文、text_spans 为文章正文经过切词和词性标注后的结果。

其中,“text_spans”和“title_spans”两列的类型为 List<Span>。Span 的定义见表 2。

每一个 Span 都是文章在分词之后的一块内容,每个 Span 包含表 2 所示的 4 个属性。

其中,word 是 Span 的词语;pos 是 Span 的词性;start 是 Span 在文章中的起始位置;end 是 Span 在文章中的结束位置。

表 1 数据模型

Tab. 1 Data model

字段	描述	类型	备注
id	自增主键	Integer	主键
title	文章标题	String	
url	文章 url	String	
date	文章发布日期	Date	
location	所属行政区划	String	如“湖北省”
child_location	二级行政区划	String	如“武汉市”
text	文章正文	Text	
title_spans	标题切词及标注后结果	List<Span>	Token 对象
text_spans	正文切词及标注后结果	List<Span>	Token 对象

表 2 Span 定义

Tab. 2 Definition of Span

字段	描述	类型	备注
word	词语	String	
pos	词性	String	如“ns”代表地名
start	在文章中的起始位置	Integer	
end	在文章中的结束位置	Integer	

本文将每篇文章的 text_spans 字段看作一张表,表中的每一行都为一个 span。这样的表示方式对于理解之后定义的诸多基于 span 之上的操作是有帮助的。具体表示方式如图 2 所示。

文章

id	title	date	location	text	spans
0001	湖南衡阳女子5年第5次离婚案宣判	2021/04/30	湖南	4月30日,宁顺花诉陈定华离婚纠纷案在湖南省衡阳市	[{湖南省,ns,18,21},{衡阳市,ns,21,24}, ...]
0002	6人非法狩猎黄鼠狼被抓:查获123张皮毛	2020/12/31	江苏	近日,江苏盐城。阮某祥常昼伏夜出,后院疑似晒出黄鼠狼	[{江苏,ns,4,5},{盐城,ns,5,6}, ...]
0003	昆明市中院维持李心草溺亡案原判,驳回上诉	2020/11/30	云南	11月30日,李心草妈妈陈美莲发微博称,她接到昆明市中级人民法院	[{昆明市,ns,23,26},{中级人民法院,n,26,32}, ...]

文章0003的spans

word	pos	start	end
昆明市	ns(地名)	23	26
中级人民法院	n(楼宇)	26	32

图 2 Spans 示例

Fig. 2 Examples of Spans

在将每篇法律文章转换为 Span 的集合之后,本文在这些 Span 上执行统一的操作来实现信息抽取。在定义完操作符后,用户可以将一系列操作符进行组合,形成一个任务,每一个任务都是对文章信息的一次抽取,可以通过一系列操作符,组合出 LDA 模型的构建过程。表 3 中列出了本文设计的主要文章

级别的操作符及其作用。

表 3　文章级别的操作符列表

Tab. 3　Article-level operators

操作符	功　能
Project(columns)	选取若干列
Select(column,value)	选取满足条件的行
Aggregate (column, func, params)	将某一列拥有相同值的行聚成一类,并应用汇总函数
Join(table,column)	将两张表合成一张
Update(column,func,params)	按照某一规则对某一列的值进行更新
Exclude(table,column)	排除两张表中共有元素的行
AddColumn(column,type)	添加一个新的列

基于上述操作符,可构建相应的信息抽取任务。给定主题数 $K = 100$、$alpha = 0.5$、$beta = 0.01$、$N = 100$,利用本文设计的操作符,实现 LDA 模型训练的过程如下:

第一步:用 Exclude 去除 Corpus 中的停用词;使用 Project 将不用的列删除;使用 AddColumn 和 Update 添加“topic”属性并且赋初值。

第二步:进行吉布斯采样,根据其采样结果,以服从多项式分布的方式,重新随机分配每个词对应的主题。为了让模型收敛,此步骤需要执行 N 次循环。

第三步:使用 Aggregate 统计每个主题的词分布。

1.2　数据处理及主题立方体

1.2.1　基于规则的行政区划归类方法

目前,多数采用机器学习或者深度学习的方法,对文章主题进行分类。基于学习的文本分类方法,不适用于本文期望实现的行政区划归类功能。由于基于学习的文本分类方法,需要大量的人工标注数据集,而本文为 OLAP 系统,要求能实现上卷、下钻等用户需求。单一粒度的地理维度显然很难满足 OLAP 用户的需求,当用户对某一主题感兴趣的时候,往往希望可以获得更细粒度的信息。

综上,本文设计了一个基于规则的行政区划归类方法。该方法首先对文本分词后的结果进行地名实体提取,再通过与三级行政区划表的比较,结合多条规则,求出最符合文章的行政区划,对于无法通过前述方法求出行政区划的文章,再利用百度地图 API 进行二次归类,最终达到一个较高的准确率。

基础算法通过比较词频的方式来决定行政区划的可信度。具体方法如下:

(1)将分词结果中词性为“ns”(地名)的词汇取出,将该词与出现次数保存进 P。

(2)遍历 P 中的地名词汇 w,将 w 与一、二、三级行政区划表中的地名进行匹配。将匹配到的行政区划 a 和 w 的加权词频存入 U 中。

(3)将 U 中每个行政区划的词频进行开根号处理(目的是降低词频饱和度),获取初步的可信度分数。

(4)对 U 中二、三级行政区划的分数加到上级行政区划上。

(5)在 U 中寻找分值最大的一级行政区划,然后在该区划中寻找分值最大的二级行政区划。

当一篇法律文章中同时出现上下两级行政区划(如:同时出现“上海市”“浦东新区”)时,这篇文章可能比只出现“上海”的文章更有可能属于上海市,即“上海市”和“浦东新区”两级行政区划在相互印证。因此,本文对同时出现上下两级行政区划的情况做出加分处理,在上面基础方法的第(3)步之后,“上海市”的分数将乘以$(1+\gamma)$,γ 为奖励系数。另外,法律文章标题中也蕴含地理位置信息。例如“湖南衡阳女子,5 年第 5 次离婚案宣判”,就能明显辨别出这篇法律文章的归属地为湖南省衡阳县。因此,本文对文章标题也进行了分词与词性标注,以同样的办法应用了上一节的(1)~(4)步,获取了标题的 U,进行加权求和。然而,部分文章虽然包含了许多行政区划,但实际上是一篇综合性的文章,无法进行明确的行政区划归类。本文将出现大于等于 4 个省的文章,当做综合性文章;将某省内大于等于 4 个市的文章当做该省内的综合性文章。综合这 3 条优化规则后,最终算法流程如算法 1 所示。

算法 1　行政区划归类算法

输入　P_{text},P_{topic} AdministrativeDivisionDictionary as dic

输出　location,child_location

for (w,tf) in P_{text},P_{topic}:

locationList = findAdministrativeDivisionInDic(w, dic)

if locationList ! = null:

for loc inlocationList:

U.put(loc,U.getOrDefault(loc, 0) + tf / locationList.size())

end for

for (loc,score) in U_{text},U_{topic}:

score = $\sqrt{score}$

```
    if haschildLoc in U:
        score = score * (1 + γ)
childLoc.score = childloc.score * (1 + γ)
for (loc,score) in Utext ,Utopic:
if loc is childLoc:
parentLoc.score += loc.score
for (loc,score) in Utext ,Utopic:
  if loc inUtext = loc in Utopic :
    score =scoretext +scoretopic
Uall.put(loc,score)
find the loc with biggest score inUall
find thechildLoc of loc with biggest socre in Uall
returnloc,childLoc
```

最后,本文对于行政区划归类方法进行了再度的优化,对那些无法进行归类的文章,利用百度地图 API 的地点检索服务。

1.2.2 主题立方体构建方法

通过对法律文章预处理和行政区划归类,法律文章拥有了时间和地点两个维度,因此主题立方体[14]可以表示为如图 3 的形式。其中横坐标为地点,纵坐标为时间,每个小方块中的内容即主题立方体的度量,即某一时间某一地点的主题分布。对于拥有时间和地点信息的法律文章数据,用户通过数据立方体,可以从时间和地域两个维度对这批数据进行 OLAP 操作。当用户想获取 2 月份上海市的法律文章时,只需要数据立方体中时间维从 2 月 1 日~2 月 29 日以及地域维为上海的数据即可。

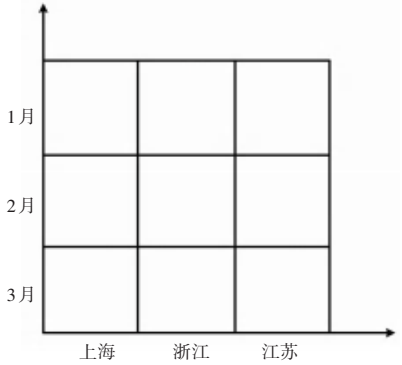

图 3 主题立方体表示方式一

Fig. 3 Representation of topic cube

由于图 3 所示的定义中缺失城市地理位置信息这一重要信息,从图中无法知悉上海市和江苏省是否相邻,这种相邻关系在计算最大独特主题范围时非常重要。因此,本文将主题立方体表示为图 4 所示的形式,更能反映主题立方体真实的状态。图 4 是一张三维的中国地图,其厚度代表时间维度,时间沿着 z 轴的方向延伸。这样的表示方式保留了各省份之间的地理位置相对关系,可以承载更多的地理位置信息,同时也更加直观,便于理解。

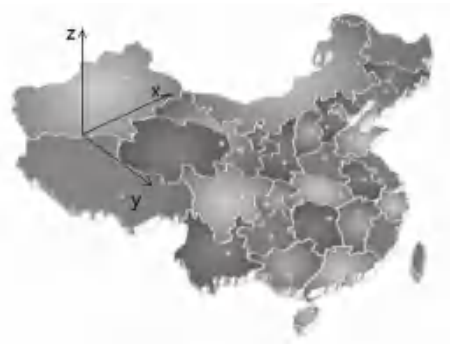

图 4 主题立方体表示方式二

Fig. 4 Another representation of topic cube

因此,本文提出了一种在多维文本集的主题立方体度量计算方法。其计算流程如下:

(1)对全局文章进行 LDA 训练:指定全局主题数量为 K_g,利用吉布斯采样的方法训练出 K_g 个全局主题 φ_g。其中,$\varphi^g = \{\varphi_1^g,\ldots,\varphi_{Kg}^g\}$。

(2)对 Cube 中的每个中等大小单元进行训练:中等大小单元为一个二维的范围。例如"上海市 1 月份"的所有法律文章,粒度选择为某省某一个月份的所有法律文章。对于每个子单元 c_i,指定主题数量为 K_l,利用吉布斯采样的方法训练出 K_l 个子主题。所以,总的子主题数 $|cell| * K_l$,$|cell|$ 为全部子单元个数。

(3)将每个步骤(2)中训练的子主题与全局主题 $\varphi^g = \{\varphi_1^g,\ldots,\varphi_{Kg}^g\}$ 进行相似度比较,利用对称 K_l 散度计算两个主题之间的距离。如果某个子主题与所有全局主题的距离都大于一个阈值 τ,说明这个子主题与所有全局主题都不相似,是一个新的主题,则将这个主题加入全局主题中;若某个子主题和某个全局主题的 K_l 散度距离很小,说明全局主题中已经包含了该子主题,则不用将这个子主题加入全局主题。

(4)经过第(3)步之后,全局主题得到了扩展,这其中的每个主题之间的 K_l 散度都超过阈值 τ,则将这些主题做为最终的主题模型 φ。最后,使用该主题模型 φ,通过吉布斯采样的方式,计算每篇文章的主题分布 θ,这一步被称为推导。

(5)获得了每篇文章的主题分布 θ 后,可以简单的聚合出每个最小单元的主题分布。此外,在时间维和地点维的不同粒度上进行聚合,以此来满足 OLAP 的上卷、下钻、切片等要求。

此后，将每篇文章的主题分布、一些单元的主题分布以及主题模型中每个主题的词概率分布存入 HBase 中，供 OLAP 模块使用。

1.3　线上查询技术

1.3.1　基于倒排索引的关键词搜索法

由于关键词搜索是线上模块，对于线上模块来说一个非常重要的指标就是时间效率。因此，OLAP 系统对线上查询的性能要求较高，找到一种提升运行效率的算法尤为关键。

本文关键词搜索算法基于倒排索引的思想，将词汇表中的所有词作为关键，将包含这个词的所有主题以及这个词在这些主题中的概率作为值，存入 HBase 中。当用户输入某几个关键词时，只需在 HBase 中找到相应的条目，统计 value 中存在的主题分数即可。例如：表 4 展示了 HBase 中有“交通肇事”、“驾驶”、“离婚”、“财产分割”等关键词，其中关键词“交通肇事”在主题 1、5、7 中出现，概率分别是 0.032 3、0.012 3、0.020 4。输入关键词“交通肇事驾驶”进行搜索，通过倒排索引可以快速查询到包含“交通肇事”和“驾驶”关键词的主题及对应的概率，对主题进行分值计算就能获得最终结果。此例中，最符合这两个关键词的主题是主题 1，其分值为 0.032 3+0.024 5=0.056 8。通过倒排索引的方式，在主题数量十分庞大的情况下，搜索引擎可以在极短的时间内返回包含关键词的文章。

表 4　关键词搜索的倒排索引

Tab. 4　Inverted index of keyword search

关键词	在不同主题中的概率
交通肇事	Topic1:0.032 3 Topic5:0.012 3 Topic7:0.020 4
驾驶	Topic1:0.024 5 Topic5:0.002 9 Topic7:0.012 4
离婚	Topic2:0.023 4 Topic4:0.053 2
财产分割	Topic2:0.012 3 Topic4:0.034 5

1.3.2　基于剪枝的最大独特主题范围查找法

独特主题是指经常出现在文档某一范围内的主题。本文用独特主题分数（*Uscore*）来评价该主题属于某一范围的独特程度。

$$UScore(r,k)=\theta_{rk}\log\left(\frac{\theta_{rk}}{\theta_{gk}}+1\right) \tag{1}$$

其中：θ_{rk} 表示主题 k 在范围 r 上的概率，θ_{gk} 表示主题 k 在全局范围上的概率。当某个主题的独特主题分数大于阈值 λ 时，说明该主题是范围 r 上的独特主题。公式为：

$$UT(r)=\{k \mid UScore(r,k)>\lambda, k\in\{1,\dots,K\}\} \tag{2}$$

其中，λ 用来衡量一个主题在全局上的的独特程度。λ 设置得越高，则被认定为独特主题的门槛就越高。这个参数可以由用户自行设定，默认值为 0.007。

对于主题集合 T 中某个主题 k 来说，所有满足 $\exists k\in T, k\in UT(r)$ 和 $\forall r'\in S, \forall k\in T: r\in r', k\notin UT(r')$ 条件范围的 r，都是该主题的最大独特主题范围。

其中，S 为全部范围集合。当 k 是范围 r 上的独特主题，而且 k 不是任何包含 r 的范围的独特主题，那么 r 就是一个 k 的最大独特主题范围。

求最大独特主题范围的基本思路是：求出所有范围的独特主题，在其中找到包含主题 k 的范围，然后筛选出最大范围。具体流程如下：

（1）计算每个范围 r 的 UScore(r,k)。

（2）找到 UScore 大于阈值的主题，作为范围 r 的独特主题。

（3）对于某个特定的主题，找到把其作为独特主题的所有范围。

（4）对步骤（3）中求出的范围进行筛选，得到最大范围集。

在地理维度上，“上海市”是一个范围，“上海市浙江省”是一个范围，“江浙沪”也是一个范围，只要省份相邻就会形成新的范围。由于需要计算所有范围的独特主题，范围的个数非常多。因此，文献[15]中提出了在时间维度上找最大独特主题范围的剪枝算法。对于任意 r 和 r'，如果 r 和 r' 的 Uscore(r,k) 和 UScore(r',k) 都小于阈值 λ，即主题 k 既不是 r 上的独特主题也不是 r' 上的独特主题，那么 k 也不可能是 $r\cup r'$ 上的独特主题。基于上述事实，本文设计了在地理维度上的剪枝方法。对于固定的主题 k，剪枝算法先计算每个最小粒度范围的独特主题分数，然后再将相邻的范围组合，再次计算独特主题分数。如果两个子范围的独特主题分数都不满足条件，则其并集必定不满足条件。当计算出湖北省的独特主题分数之后，要找到其所有的邻居省份，分别计算取并集之后的独特主题分数。如果满足最大独特主题范围，则保留该省份。

2　系统评估

2.1　实验环境

为了验证系统的准确性以及线上即时交互的效率，对系统整体进行了测试。测试内容包括对行政区划归类模块的准确性测试和线上查询响应时间测试。

系统数据爬取自中国新闻网 2019 年 1 月 ~ 2020 年 5 月发布的法律新闻数据,共347 635篇,去重后为273 029篇。系统使用 Java 语言开发,硬件环境见表 5。

表 5 硬件环境

Tab. 5 Hardware environment

集群节点数目	3
操作系统	Ubuntu Linux 14.04 64 位
处理器	Intel Xeon CPU E5-2620 v2 @ 2.10GHz, 24 核
节点内存	32GB
节点硬盘存储	1.8TB
Spark 版本	Spark-2.3.1
Java 版本	Java-8

2.2 行政区划归类准确性测试

通过人工标注方式,对 530 篇文章进行了手工标注,标注形式为<一级行政区划,二级行政区划>。如,法律新闻《张家界通报导游怒骂游客骗吃骗喝调查处理:对旅行社罚 20 万》被标注为<湖南省,张家界市>。实验对基础方法、优化方法,以及使用百度地图 API 辅助方法进行测试,测试结果,见表 6。

表 6 行政区划归类准确性测试

Tab. 6 Experiment of the accuracy of administrative divisions

方法	时间	准确率	未找到行政区划
基于规则的行政区划归类方法	5.13 s	90.1%	2.1%
优化方法	6.25 s	93.8%	2.1%
优化方法+百度地图 API 辅助	8.92s	95.6%	0.0%

由此可见,无论是否使用地图 API,优化方法的准确率都超过了 93%,使用地图 API 辅助的方法准确率达到了 95.6%,但在时间上的劣势比较明显。主要原因是每次使用地图 API 都需要进行一次远程调用,在这次实验当中,共有 11 篇文章未在行政区划识别词典中找到相关的行政区划,这部分文章都需要使用地图 API 进一步搜索。可以看出,调用地图 API 的成本比较大,但是总体的时间复杂度以及准确率均尚可。

2.3 线上查询响应时间测试

在线上查询响应时间的测试中,经反复调整输入的时间区间以及关键词,最后得到每个小模块的响应时间,结果如图 5 所示。可以看到,关键词搜索的平均响应时间在 0.02 s 左右。最大独特主题范围在 1.2 s 左右,下钻操作本身也是一个最大独立范围查询的过程,但是因为地图的范围比较小,所以查询时间比较短。相关新闻查询的平均响应时间为 70 ms,虽然每个主题在任意空间和时间范围内的相关文章已经预先存储在 posting list 中。但因 posting list 空间占用过大,在增量压缩之后依然无法放入内存中,其时间主要消耗在读盘以及解压缩中。

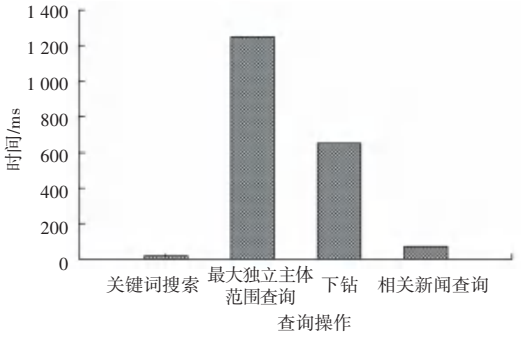

图 5 平均响应时间

Fig. 5 Average response time

3 结束语

本文设计并实现了一个面向司法大数据的文本主题 OLAP 系统,设计了新的数据模型 Span 保存数据,并针对这种新的数据模型设计了多个操作符,后续的数据处理均由这些操作符组合实现。

针对文章无地理位置维度问题,提出了基于规则的二级行政区划归类,对提取出的地名词汇进行了可信度排序,从而得到最符合文章条件的行政区划。对于无法匹配的地名,通过调用百度地图 API 的方式进行识别,识别准确率高达 95%以上。

在关键词相关主题匹配问题上,采用倒排索引的方式,将一个词语对应多个与之相关的主题,从而将单个关键词匹配主题的时间复杂度降低到常数级别。对于最大独特主题范围查询问题,本文提出一种在地理维度上快速寻找最大独特主题范围的方法。平均线上查询时间在优化过后保持在 1 s 以内。

参考文献

[1] BLEI D M, NG A Y, JORDAN M I. Latent dirichlet allocation [J]. Journal of machine Learning research, 2003, 3(1): 993-1022.

[2] FISCUS J G, DODDINGTON G R. Topic detection and tracking evaluation overview[M]//Topic detection and tracking. Springer, Boston, MA, 2002: 17-31.

[3] DOYLE G, ELKAN C. Accounting for burstiness in topic models [C]//Proceedings of the 26th Annual International Conference on Machine Learning. 2009: 281-288.

[4] YANG Z, MA H, HE Z, et al. Finding maximal ranges with unique topics in a text database[J]. World Wide Web, 2018, 21(2): 289-310.

[5] VORA M N. Hadoop - HBase for large - scale data [C]// Proceedings of 2011 International Conference on Computer Science and Network Technology. IEEE, 2011: 601-605.

(下转第 41 页)