(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 109033358 B
(45)授权公告日 2022.06.10

(21)申请号 201810832345.1

(22)申请日 2018.07.26

(65)同一申请的已公布的文献号
申请公布号 CN 109033358 A

(43)申请公布日 2018.12.18

(73)专利权人 李辰洋
地址 100024 北京市朝阳区康惠园2号院4号楼2层208

(72)发明人 李辰洋

(74)专利代理机构 北京汇信合知识产权代理有限公司 11335
专利代理师 孙民兴

(51)Int.Cl.
G06F 16/9535(2019.01)
G06F 16/9536(2019.01)
G06F 16/35(2019.01)
G06F 40/295(2020.01)

(56)对比文件
CN 106095762 A,2016.11.09
CN 102364473 A,2012.02.29
CN 105022827 A,2015.11.04
王鹏.基于新闻网页主题要素的网页去重方法研究.《计算机工程与应用》.2007,
高凯.网页去重策略.《上海交通大学学报》.2006,

审查员 王晓霞

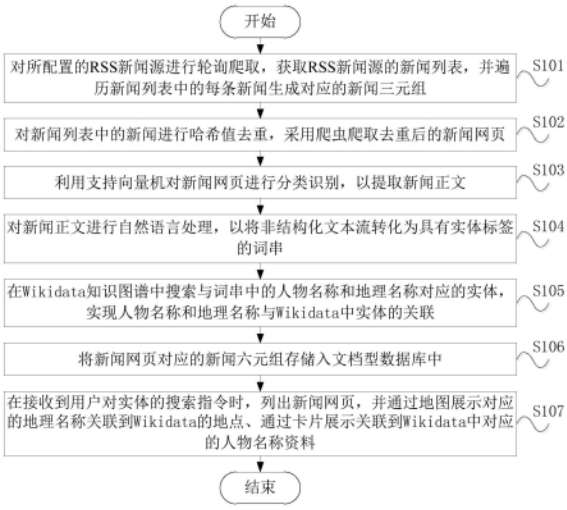
权利要求书2页 说明书9页 附图2页

(54)发明名称

新闻聚合与智能实体关联的方法

(57)摘要

本发明公开了一种新闻聚合与智能实体关联的方法,通过轮询用户感兴趣的网站上的新增新闻,采用爬虫抓取网页,并利用支持向量机进行0-1分类以提取新闻正文,对新闻正文进行自然语言处理后,对文本中出现的人物和地理名称在Wikidata知识图谱中搜索对应实体,通过上位词确定实体类型,将新闻六元组(标题,时间,URL,正文,人物实体,地理实体)存入本地文档数据库,在用户搜索相关实体时,列出相关新闻,并通过地图展示关联到Wikida的新闻地点,通过卡片展示关联到Wikida中的人物档案。通过本发明的技术方案,提供了一种关联知识推送的具有背景知识的增强型新闻阅读方式,改善了用户的阅读体验。



CN 109033358 B

1.一种新闻聚合与智能实体关联的方法,其特征在于,包括:

对所配置的RSS新闻源进行轮询爬取,获取所述RSS新闻源的新闻列表,并遍历所述新闻列表中的每条新闻生成对应的新闻三元组;

对所述新闻列表中的新闻进行哈希值去重,采用爬虫爬取去重后的新闻网页;

利用支持向量机对所述新闻网页进行分类识别,以提取新闻正文;

采用Stanford NLP自然语言处理框架对所述新闻正文进行分词、断句、词性标注、命名实体识别的自然语言处理,将非结构化文本流转化为具有实体标签的词串,以提取人物名称和地理名称;

在Wikidata知识图谱中搜索与所述词串中的人物名称和地理名称对应的实体,实现所述人物名称和所述地理名称与Wikidata中所述实体的关联,具体包括:

在Wikidata中利用HTTP API接口向Wikidata提取人物名称和地理名称对应的实体,并通过该实体的上位词进行消歧;

在所述人物名称和人物实体之间、所述地理名称和地理实体之间分别建立名称-实体的映射关系,实现所述人物名称和所述地理名称与Wikidata中对应实体的关联,其中,带有经纬信息的实体也作为地理实体;将所述新闻网页对应的新闻六元组存储入文档型数据库中;

在接收到用户对所述实体的搜索指令时,列出所述新闻网页,并通过地图展示对应的地理名称所关联的Wikidata中地点信息、通过卡片展示所述Wikidata中对应的人物名称资料,

其中,所述新闻三元组包括标题、时间和URL,所述新闻六元组包括标题、时间、URL、正文、人物实体和地理实体。

2.根据权利要求1所述的新闻聚合与智能实体关联的方法,其特征在于,所述对所述新闻列表中的新闻进行哈希值去重,采用爬虫爬取去重后的新闻网页具体包括:

计算所述新闻列表中的每条新闻对应的URL计算哈希值,查询本地爬取列表的哈希表中是否存在相同哈希值;

若本地爬取列表中不存在,则查询所述文档型数据库中是否存在该新闻,若所述文档型数据库和所述本地爬取列表中均不存在该新闻,则将该新闻插入爬取队列中进行爬取,否则处理下一条新闻。

3.根据权利要求1所述的新闻聚合与智能实体关联的方法,其特征在于,所述利用支持向量机对所述新闻网页进行分类识别,以提取新闻正文具体包括:

向该新闻的URL请求HTML格式的新闻网页,并通过网页降噪规则去除页面噪声;

利用支持向量机对去除噪声后的页面元素进行0-1分类识别,并提取新闻正文。

4.根据权利要求1所述的新闻聚合与智能实体关联的方法,其特征在于,在将所述新闻网页对应的新闻六元组存储入文档型数据库的过程中,将所述新闻网页的原始正文和转化出的具有实体标签的词串同样存入所述文档型数据库。

5.根据权利要求1所述的新闻聚合与智能实体关联的方法,其特征在于,所述地图为世界地图或局部地区地图,所述人物名称资料包括人物照片、人物姓名和人物简介。

6.根据权利要求1所述的新闻聚合与智能实体关联的方法,其特征在于,所述对新闻源的轮询爬取间隔为5分钟。

7.根据权利要求1所述的新闻聚合与智能实体关联的方法,其特征在于,在接收到用户对所述实体的搜索指令时,根据时间由新到旧的列出所述新闻网页。

8.根据权利要求1所述的新闻聚合与智能实体关联的方法,其特征在于,所述知识图谱为维基知识图谱Wikidata,所述文档型数据库选择MongoDB。

新闻聚合与智能实体关联的方法

技术领域

[0001] 本发明涉及信息检索方法技术领域，尤其涉及一种新闻聚合与智能实体关联。

背景技术

[0002] 随着互联网Web2.0、社交网络、移动互联网的发展，新闻发生到经过社交网络、门户网站、主流媒体的传播几乎成为秒级事件，特别是机器参与新闻的采集、生成和转发，导致了海量新闻充斥网络，使用户处于数据汪洋之中，难以发现有价值的新闻数据。实际上，在舆情监控领域，用户关注的是与自身密切相关的主题和关键词的新闻传播与事件影响力。对于普通用户，希望通过聚合新闻，了解天下大事，需要读取新闻的同时了解相关的新闻发生地理信息和人物信息，以洞悉新闻事件的背景资料和关联知识。因此，通过知识图谱实现文本的智能实体标注提供有背景知识的新闻成为一种带有普遍性的用户需求。

[0003] (1)国内著名的新闻聚合网站有百度新闻、今日头条、UC头条、天天快报、电力头条等。这些网站通过爬虫聚合全网新闻数据，通过算法和人工推荐，实现用户的定制化新闻阅读，提高信息获取效率。该方法存在对用户个体兴趣和群体点击的过拟合，导致推荐有效性不高，存在泛娱乐化问题。此外，这些方法仅提供了新闻正文，没能有效利用新闻背景信息进行信息增强和可视化展示。

[0004] (2)带有噪声抑制的主题爬虫研究概况。2014年斯坦福大学的Ziyan Zhou等采用DOM树标签、CSS样式和页面元素几何特征输入SVM分类器识别网页正文。2015年，Mozilla公司的Matthew E.Peters等人采用页面元素的文本统计特征进行线性分类，达到了商业产品级的使用性能，并作为新功能嵌入了Mozilla公司的Firefox浏览器。

[0005] 支持向量机(SVM)基于结构风险最小化理论在特征空间中构建最优超平面，使得学习器得到全局最优化。支持向量机属于统计学习方法，建立在坚实的理论基础上，有着不需要特定领域的专业知识、易于迁移、适合高维数据的处理、能解决小样本问题、泛化性能较好等优点，在文本分类、图像识别等分类问题中有良好的表现。

[0006] 实际上，正文提取就是在XML/HTML上的文本分类，通常正文文本HTML元素具有段落元素多、元素样式类中包含类似“content”“body”关键词、页面几何占比大的特征。Christian Kohlschütter等开发的Boilerpipe正文提取框架，基于SVM提取正文，并提供API。

[0007] (3)命名实体识别技术。斯坦福大学自然语言处理组的Jenny Rose Finkel等采用具有全局特征的条件随机场(CRF)实现了命名实体识别，具有业界领先的识别性能。

[0008] 国内，杨东华等在大数据清洗过程优化中计算实体相似度，采用并行实体聚类，实现实体识别。王宏志李亚坤等研究了数据质量管理中的实体识别，用于错误检测、不一致数据发现等，将传统文本实体识别推广到XML数据、图数据和复杂网络上。孙琛琛等研究了面向关联数据的联合式实体识别，将相似度算法应用在对象图上，迭代地收缩相似节点，实现实体聚类。寇月等利用关联实体识别技术对异构网络中主题相关的实体检测并整合，更好地帮助用户理解搜索目标。高俊平等基于条件随机场研究了面向中文维基百科领域知识的

演化关系抽取方法，利用语法分析特征，挖掘演化关系模式，构建演化关系推理模型。

[0009] (4)知识图谱技术。2007年，美国公司Metaweb创立了开放知识图谱Freebase，其采用实体关系模型，通过维基百科词条生成高度结构化的数据，后被Google收购，一度成为世界上最大的知识图谱，但该项目在2014年停止运营。2012年，Wikimedia基金会创立了Wikidata(维基数据)计划，透过与维基百科的开放互动接口，实现维基百科半结构化数据的结构化重建，是目前全世界最大的开放知识图谱。Wikidata是基于群智完成的知识图谱，错误率较低，且提供了易用API，目前包含5100万实体。Wikidata全部数据可以下载，以CC0协议发布，放弃著作权，允许复制、修改、发行和演绎，属于公共领域知识图谱。国内百度基于搜索引擎的大数据，建立了知识图谱，并应用于智能问答、实体推荐、对话系统和智能客服。

[0010] (5)MongoDB文档数据库。MongoDB是由MongoDB.Inc开发的基于JSON的文档数据库，相较传统RDBMS，MongoDB具有无模型、半结构化的特点，更适合新闻文本存储任务。

[0011] (6)地理信息可视化技术。D3.JS是全世界最著名的开源可视化工具包，通过TopoJSON传入地图边界数据，标注地理位置。ECharts是百度开源的数据可视化工具包，也可以完成上述功能。

发明内容

[0012] 针对上述问题中的至少之一，本发明提供了一种新闻聚合与智能实体关联的方法，通过轮询用户感兴趣的网站上的新增新闻，采用爬虫抓取网页，并利用支持向量机进行0-1分类以提取新闻正文，对新闻正文进行自然语言处理后，对文本中出现的人物和地理名称在Wikidata知识图谱中搜索对应实体，通过上位词确定实体类型，将新闻六元组(标题，时间，URL，正文，人物实体，地理实体)存入数据库，在用户搜索相关实体时，列出相关新闻，并通过地图展示关联到Wikidata的新闻地点，通过卡片展示关联到Wikidata的人物档案，提供了一种关联知识推送的具有背景知识的增强型新闻阅读方式。

[0013] 为实现上述目的，本发明提供了一种新闻聚合与智能实体关联的方法，包括：对所配置的RSS新闻源进行轮询爬取，获取所述RSS新闻源的新闻列表，并遍历所述新闻列表中的每条新闻生成对应的新闻三元组；对所述新闻列表中的新闻进行哈希值去重，采用爬虫爬取去重后的新闻网页；利用支持向量机对所述新闻网页进行分类识别，以提取新闻正文；对所述新闻正文进行自然语言处理，以将非结构化文本流转化为具有实体标签的词串；在Wikidata知识图谱中搜索与所述词串中的人物名称和地理名称对应的实体，实现所述人物名称和所述地理名称与Wikidata中所述实体的关联；将所述新闻网页对应的新闻六元组存储入文档型数据库中；在接收到用户对所述实体的搜索指令时，列出所述新闻网页，并通过地图展示对应的地理名称关联的Wikidata中地点信息、通过卡片展示所述Wikidata中对应的人物名称资料，其中，所述新闻三元组包括标题、时间和URL，所述新闻六元组包括标题、时间、URL、正文、人物实体和地理实体。

[0014] 在上述技术方案中，优选地，所述对所述新闻列表中的新闻进行哈希值去重，采用爬虫爬取去重后的新闻网页具体包括：计算所述新闻列表中的每条新闻对应的URL计算哈希值，查询本地爬取列表的哈希表中是否存在相同哈希值；若本地爬取列表中不存在，则查询所述文档型数据库中是否存在该新闻，若所述文档型数据库和所述本地爬取列表中均不

存在该新闻，则将该新闻插入爬取队列中进行爬取，否则处理下一条新闻。

[0015] 在上述技术方案中，优选地，所述利用支持向量机对所述新闻网页进行分类识别，以提取新闻正文具体包括：向该新闻的URL请求HTML格式的新闻网页，并通过网页降噪规则去除页面噪声；利用支持向量机对去除噪声后的页面元素进行0-1分类识别，并提取新闻正文。

[0016] 在上述技术方案中，优选地，所述对所述新闻正文进行自然语言处理，以将非结构化文本流转化为具有实体标签的词串具体包括：采用Stanford NLP自然语言处理框架对所述新闻正文进行分词、断句、词性标注、命名实体识别，以提取人物名称和地理名称。

[0017] 在上述技术方案中，优选地，所述在Wikidata知识图谱中搜索与所述词串中的人物名称和地理名称对应的实体，实现所述人物名称和所述地理名称与Wikidata中所述实体的关联具体包括：在Wikidata中利用HTTP API接口向Wikidata提取人物名称和地理名称对应的实体，并通过该实体的上位词进行消歧；在所述人物名称和人物实体之间、所述地理名称和地理实体之间分别建立映射关系，实现所述人物名称和所述地理名称与Wikidata中对应实体的关联，其中，带有经纬信息的实体也作为地理实体。

[0018] 在上述技术方案中，优选地，在将所述新闻网页对应的新闻六元组存储入文档型数据库的过程中，将所述新闻网页的原始正文和转化出的具有实体标签的词串同样存入所述文档型数据库。

[0019] 在上述技术方案中，优选地，所述地图为世界地图或局部地区地图，所述人物名称资料包括人物照片、人物姓名和人物简介。

[0020] 在上述技术方案中，优选地，所述对新闻源的轮询爬取间隔为5分钟。

[0021] 在上述技术方案中，优选地，在接收到用户对所述实体的搜索指令时，根据时间由新到旧的列出所述新闻网页。

[0022] 在上述技术方案中，优选地，所述知识图谱为维基知识图谱Wikidata，所述文档型数据库选择MongoDB。

[0023] 与现有技术相比，本发明的有益效果为：通过轮询用户感兴趣的网站上的新增新闻，采用爬虫抓取网页，并利用支持向量机进行0-1分类以提取新闻正文，对新闻正文进行自然语言处理后，对文本中出现的人物和地理名称在Wikidata知识图谱中搜索对应实体，通过上位词确定实体类型，将新闻六元组(标题，时间，URL，正文，人物实体，地理实体)存入数据库，在用户搜索相关实体时，列出相关新闻，并通过地图展示新闻地点，通过卡片展示关联到的人物档案，提供了一种关联知识推送的具有背景知识的增强型新闻阅读方式。

附图说明

[0024] 图1为本发明一种实施例公开的新闻聚合与智能实体关联的方法的流程示意图；

[0025] 图2为本发明一种实施例公开的新闻聚合与智能实体关联的部署环境示意图。

具体实施方式

[0026] 为使本发明实施例的目的、技术方案和优点更加清楚，下面将结合本发明实施例中的附图，对本发明实施例中的技术方案进行清楚、完整地描述，显然，所描述的实施例是本发明的一部分实施例，而不是全部的实施例。基于本发明中的实施例，本领域普通技术人

员在没有做出创造性劳动的前提下所获得的所有其他实施例，都属于本发明保护的范围。

[0027] 下面结合附图对本发明做进一步的详细描述：

[0028] 如图1所示，根据本发明提供的一种新闻聚合与智能实体关联的方法，包括：步骤S101，对所配置的RSS新闻源进行轮询爬取，获取RSS新闻源的新闻列表，并遍历新闻列表中的每条新闻生成对应的新闻三元组；步骤S102，对新闻列表中的新闻进行哈希值去重，采用爬虫爬取去重后的新闻网页；步骤S103，利用支持向量机对新闻网页进行分类识别，以提取新闻正文；步骤S104，对新闻正文进行自然语言处理，以将非结构化文本流转化为具有实体标签的词串；步骤S105，在Wikidata知识图谱中搜索与词串中的人物名称和地理名称对应的实体，实现人物名称和地理名称与Wikidata中实体的关联；步骤S106，将新闻网页对应的新闻六元组存储入文档型数据库中；步骤S107，在接收到用户对实体的搜索指令时，列出新闻网页，并通过地图展示对应的地理名称关联到Wikidata的地点信息、通过卡片展示Wikidata中对应的人物名称资料，其中，新闻三元组包括标题、时间和URL，新闻六元组包括标题、时间、URL、正文、人物实体和地理实体。

[0029] 在该实施例中，根据用户兴趣，配置新闻网站来源。配置支持RSS的新闻网站列表，默认配置腾讯新闻、百度新闻、新浪新闻，也可配置其他支持RSS格式的新闻源，通过轮询RSS新闻源，计算新闻网址的哈希值去重复，采用爬虫增量抓取新闻网页。其中，对RSS新闻源的轮询时间优选为5分钟。

[0030] 具体地，爬虫模块基于Java实现，数据库采用MongoDB，采用Node.JS实现了HTTP面向前端的接口，前端采用Facebook React和D3.JS实现交互界面和可视化，本发明提出的方法搭建的系统部署在微软WindowsAzure服务器上，访问地址http://analy.news/。系统爬虫模块包括：5个包、15个类、7个内部依赖关系。

[0031] 爬虫模块的配置采用YAML(YAML Ain't Markup Language)格式，其中各配置项为：Delay：配置项指两次爬取的休息时间，以毫秒为单位，默认五分钟；Concurrent：配置项指自然语言分析器线程池的并行线程数，默认为4个线程；MongoUri和database：配置项指MongoDB数据库的地址和数据库名，数据库默认端口为27017，采用了本地部署的数据库；Cache：配置项是本地最近爬取列表的长度，默认为1024，该值大小应该略大于所有RSS新闻源新闻列表的总长，减少缓存未命中情况下的数据库查询次数；Feeds：配置项为RSS新闻源列表，每个新闻源有三个属性：name名称、url新闻源地址、lang新闻源语言，新闻源语言代码遵循RFC5646规范，如简体中文为zh-Hans。

[0032] 爬虫模块的最外层布局包括crawler、analyzer、model、mongo、tagger五个包、ConfigManager类和config.Config一个依赖类。其中，Crawler包实现了爬虫主函数；Analyzer包实现了新闻的正文提取、去重、分析、入库工作流；Model包包含了所有爬虫用到的所有POJO格式(Plain Ordinary Java Object，简单的Java对象)数据模型类；Mongo包实现了MongoDB数据库增删改查辅助代码；Tagger包实现了自然语言处理、实体关联主要算法。

[0033] ConfigManager为配置管理器类，实现了YAML格式爬虫配置的读取，通过单例模式向其他程序提供程序运行过程中唯一的model.Config外部类实例。crawler包内部布局中，共有2个类，2个依赖类。Crawler类为整个项目的主类，包含一个入口点程序，引导读取配置文件、连接数据库、配置定时爬取任务，其依赖analyzer.Analyzer和mongo.MongoManager

两个类。CrawlerJob类实现了定时轮询新闻源，并把所有新闻放入任务队列，供分析器处理。

[0034] MongoDB数据库模型中，model包内部布局共有6个类。News类为单条新闻的数据模型，包括数据库id、基于url哈希的全局唯一标识uuid、新闻url、新闻标题title、新闻纯文本正文content、新闻语言lang、新闻发布日期pubDate、经过自然语言处理标注的新闻正文tagged、正文中出现的地理实体gpeTag、正文中出现的人物实体personTag。

[0035] 配置管理的各个数据模型中：Config类为配置文件数据模型；Feed类为配置文件中单个新闻源的数据模型，包括名称name，新闻源RSS端点url，新闻源语言lang；GeoEntity类为地理实体数据模型，包括该实体的Wikidata知识图谱唯一标识id、经度longitude、维度latitude、别名names、出现次数hits；PersonEntity类为人物实体数据模型，包括该实体的Wikidata知识图谱唯一标识id、别名names、出现次数hits；Term类为经过自然语言处理标注的新闻中单个词的数据模型，包括词组n、识别出的实体类型t。

[0036] 在该实施例中，通过HTTP协议向当前RSS源请求RSS XML，该XML描述了该新闻源最近发布的新闻列表。遍历新闻列表，对每条新闻生成新闻三元组(标题，时间，URL)。对于当前RSS源中的每条新闻依次处理，该源中新闻处理完毕后，请求下一个RSS源。

[0037] 在上述实施例中，优选地，对新闻列表中的新闻进行哈希值去重，采用爬虫爬取去重后的新闻网页具体包括：计算新闻列表中的每条新闻对应的URL计算哈希值，查询本地爬取列表的哈希表中是否存在相同哈希值；若本地爬取列表中不存在，则查询文档型数据库中是否存在该新闻，若文档型数据库和本地爬取列表中均不存在该新闻，则将该新闻插入爬取队列中进行爬取，否则处理下一条新闻。

[0038] 其中，具体的，新闻列表中的新闻可能是之前已经抓取过的，对三元组中URL计算哈希，利用哈希值在本地爬取列表构成的哈希表中寻找对应项目，若本地列表中不存在该新闻，则向MongoDB发起查询，请求数据库中是否存在该新闻。若该新闻已被爬取过，则丢弃并处理下一条新闻；若没有爬取过，插入分析队列，进行后续分析。利用哈希表缓存，显著减少了数据库去重查询，降低了后端数据库负载，从而提高了去重效率。

[0039] mongo包内部仅有1个类。MongoManager类通过MongoDB的JavaAPI实现了对MongoDB数据库的操作，包括连接数据库、checkExist查询某条新闻是否已收录、insertNews插入一条新闻。对于查询收录，实现了基于LRU(Least recently used，最近最少使用)的本地缓存以提高查询性能。

[0040] 在上述实施例中，优选地，利用支持向量机对新闻网页进行分类识别，以提取新闻正文具体包括：向该新闻的URL请求HTML格式的新闻网页，并通过Adblock提供的网页降噪规则去除页面中广告、视频、动态图片、Flash控件、Java Applet控件等页面噪声；将剩余页面元素输入SVM分类器，利用支持向量机对去除噪声后的页面元素进行0-1分类识别，并提取新闻正文，形成新闻四元组(标题，时间，URL，正文)。

[0041] 在上述实施例中，优选地，对新闻正文进行自然语言处理，以将非结构化文本流转化为具有实体标签的词串具体包括：采用Stanford NLP自然语言处理框架对新闻正文进行分词、断句、词性标注、命名实体识别，以提取人物名称和地理名称，将非结构化文本流转化为有实体标签的词串。

[0042] 例如：

[0043] 当地时间5月5日,中国赠送马克思雕像在德国揭幕。

[0044] 经分词、断句、词性标注、命名实体识别后得到:

[0045] 当地/时间/5月/5日/,/中国/赠送/马克思/雕像/在/德国/揭幕/。

[0046] analyzer包的内部共有2个类,1个依赖类。Analyzer类用于接收上游Crawler产生的单条新闻任务,加入到任务队列中,并通过AnalyzerJob类描述的过程进行处理。AnalyzerJob类调用MongoManager类实现了新闻去重,调用Boilerpipe正文提取库实现了正文提取,调用tagger包下的各个标注算法实现了对单条新闻的分析,最后将新闻插入数据库。

[0047] 在上述实施例中,优选地,在Wikidata知识图谱中搜索与词串中的人物名称和地理名称对应的实体,实现人物名称和地理名称与Wikidata中实体的关联具体包括:在Wikidata中利用HTTP API接口向Wikidata提取人物名称和地理名称对应的实体,并通过该实体的上位词进行消歧;在人物名称和人物实体之间、地理名称和地理实体之间分别建立映射关系,实现人物名称和地理名称与Wikidata中对应实体的关联,其中,带有经纬信息的实体也作为地理实体。

[0048] 具体地,知识图谱为维基数据Wikidata时,利用Wikidata知识图谱,完成人物、地理实体消歧,建立名称-实体的映射。对于词串中识别出的人名、地名,利用Wikidata HTTP API向知识图谱发起请求,搜索该名称相关的实体。对于识别出的人名,从返回的实体列表中寻找具有上位词“人”的实体;对于识别出的地名,从返回的实体列表中寻找具有上位词“地点”或具有经纬度数据的实体。

[0049] 这种简单实体消歧,可以通过知识图谱上位词分类的方法,使人名“马克思”不会识别成抽象实体“马克思主义”。构造名称-实体映射,形成新闻六元组(标题,时间,URL,正文,人物列表,地理列表),存入MongoDB数据库。表1展示了一个短新闻的数据库条目。

[0050] 表1存储在MongoDB中的新闻六元组

[0051]


```
{
    "_id":"5aeda4eb93020609c51f7ef2",
    "content":"当地时间 5 月 5 日，中国赠送马克思雕像在德国揭幕。",
    "gpeTag":[
      {
      "_id":"Q148",
      "hits":1,
      "latitude":35,
      "longitude":103,
      "names":["中国"]
      },
      {
      "_id":"Q183",
      "hits":1,
      "latitude":51,
      "longitude":10,
      "names":["德国"]
      }
      ],
    "lang":"zh-Hans",
    "personTag":[
      {
      "_id":"Q9061",
      "hits":1,
      "names":["马克思"]
      }
      ],
    "pubDate":"2018-05-05T11:53:08.000Z",
    "tagged":[
      {"n":"当地","t":"O"},
      {"n":"时间","t":"O"},
      {"n":"5 月","t":"MISC"},
```


[0052]


```
      {"n":"5 日","t":"MISC"},
      {"n":"中国","t":"GPE"},
      {"n":"赠送","t":"O"},
      {"n":"马克思","t":"PERSON"},
      {"n":"雕像","t":"O"},
      {"n":"在","t":"O"},
      {"n":"德国","t":"GPE"},
      {"n":"揭幕","t":"O"}
    ],
    "title":"中国赠送马克思雕像在德国揭幕(图)|马克思|雕像|德国_新浪新闻",
    "url":"http://news.sina.com.cn/c/2018-05-05/doc-ifyuwqfa6880785.shtml
",
    "uuid":"941a842ac5a5d83a6a324422b03ff83fd266aa064dee37eaf43379b90ed62
8a4"
}
```


[0053] tagger包用于实现NLP与智能实体关联相关算法，包括3个类。NERTagger类调用StanfordNLP包的API实现了对新闻正文的分词、断句、词性标注、命名实体识别。PersonMapper类调用Wikidata知识图谱API实现了对新闻正文人物实体的智能消歧。GPEMapper类调用Wikidata知识图谱API实现对新闻正文地理实体的智能消歧。

[0054] Wikidata的HTTP API入口为:https://www.wikidata.org/w/api.php，以“马克思”为例，如果要访问“马克思”的全部实体信息，其在Wikidata中的实体ID为Q9061，访问：

api.php?action＝wbgetclaims&entity＝Q9061&format＝json,会返回一个JSON文档。

[0055] Wikidata的知识图谱模型分为实体和属性两类,每个实体具有一个以Q开头的编码,如“马克思”的实体编码为Q9061,“德国”为Q183,每个属性具有一个以P开头的编码,如“性质”属性为P31,“地理坐标”属性为P625。

[0056] 在该实施例中,优选地,可采用Facebook的React前端框架和D3.JS实现新闻展示。React视图采用包含以自定义HTML标记规定的其他组件来渲染组件。React提供了一种子组件不能直接影响外层组件“data flows down”的模型,数据改变时对HTML文档及时更新,实现与单页应用中组件之间干净分离。系统前端采用Facebook React和D3.JS实现。表2展示了项目前端的依赖,bootstrap为前端UI框架、D3为前端可视化框架、fetch-jsonp用于使FetchAPI兼容JSONP请求规范、leaflet用于展示用户可读的时间。

[0057] 表2一种新闻聚合与智能实体关联方法的前端依赖软件列表

[0058]

```
{
    "dependencies":{
        "bootstrap":"^3",
        "d3":"^4.13.0",
        "fetch-jsonp":"^1.1.3",
        "leaflet":"^1.3.1",
        "react":"^16.2.0"
    },
    "devDependencies":{
        "babel-preset-es2015":"^6.24.1",
        "babel-preset-react-app":"^3.1.1",
        "babel-runtime":"^6.23.0"
    }
}
```

[0059] 通过W3C规定的HTML5下一代资源获取接口Fetch API,浏览器端的代码向Node.JS实现的REST API发起请求,获取数据库中对应的新闻条目。在前端通过fetch-jsonp库实现与Wikidata HTTP API的基于JSONP通信规范的跨域通信,向Wikidata请求新闻中实体的图片和属性描述。

[0060] 如图2所示为本发明提供的新闻聚合与智能实体关联的方法的部署的软件栈,优选地,本方法实现的系统部署在云服务器上,双核CPU,内存2.5G,安装Ubuntu 16.04LTS操作系统,绑定域名analy.news,需要安装的软件包括:MongoDB3.6、OpenJDK8、Node.JS 9、Lighttpd1.4。

[0061] 优选地,用户访问网站时,首屏默认随机展示一条新闻的正文、地理分布和人物卡片。用户搜索人名、地名时,利用Wikidata HTTP API向知识图谱发起请求,搜索该名称相关的实体,并从数据库中取出相关新闻。

[0062] 在上述实施例中,优选地,在将新闻网页对应的新闻六元组存储入MongoDB文档型数据库的过程中,将新闻网页的原始正文和转化出的具有实体标签的词串同样存入MongoDB文档型数据库。

[0063] 在上述实施例中,优选地,地图为世界地图或局部地区地图,人物名称资料包括人物照片、人物姓名和人物简介。

[0064] 在上述实施例中,优选地,对新闻源的轮询爬取间隔为5分钟。

[0065] 在上述实施例中,优选地,在接收到用户对实体的搜索指令时,根据时间由新到旧

的列出新闻网页。

[0066] 以上所述为本发明的实施方式，根据本发明提出的新闻聚合与智能实体关联的方法，通过轮询用户感兴趣的网站上的新增新闻，采用爬虫抓取网页，并利用支持向量机进行0-1分类以提取新闻正文，对新闻正文进行自然语言处理后，对文本中出现的人物和地理名称在Wikidata知识图谱中搜索对应实体，通过上位词确定实体类型，将新闻六元组(标题，时间，URL，正文，人物实体，地理实体)存入数据库，在用户搜索相关实体时，列出相关新闻，并通过地图展示新闻地点，通过卡片展示关联到的人物档案，提供了一种关联知识推送的具有背景知识的增强型新闻阅读方式，改善了用户的阅读体验。

[0067] 以上仅为本发明的优选实施例而已，并不用于限制本发明，对于本领域的技术人员来说，本发明可以有各种更改和变化。凡在本发明的精神和原则之内，所作的任何修改、等同替换、改进等，均应包含在本发明的保护范围之内。

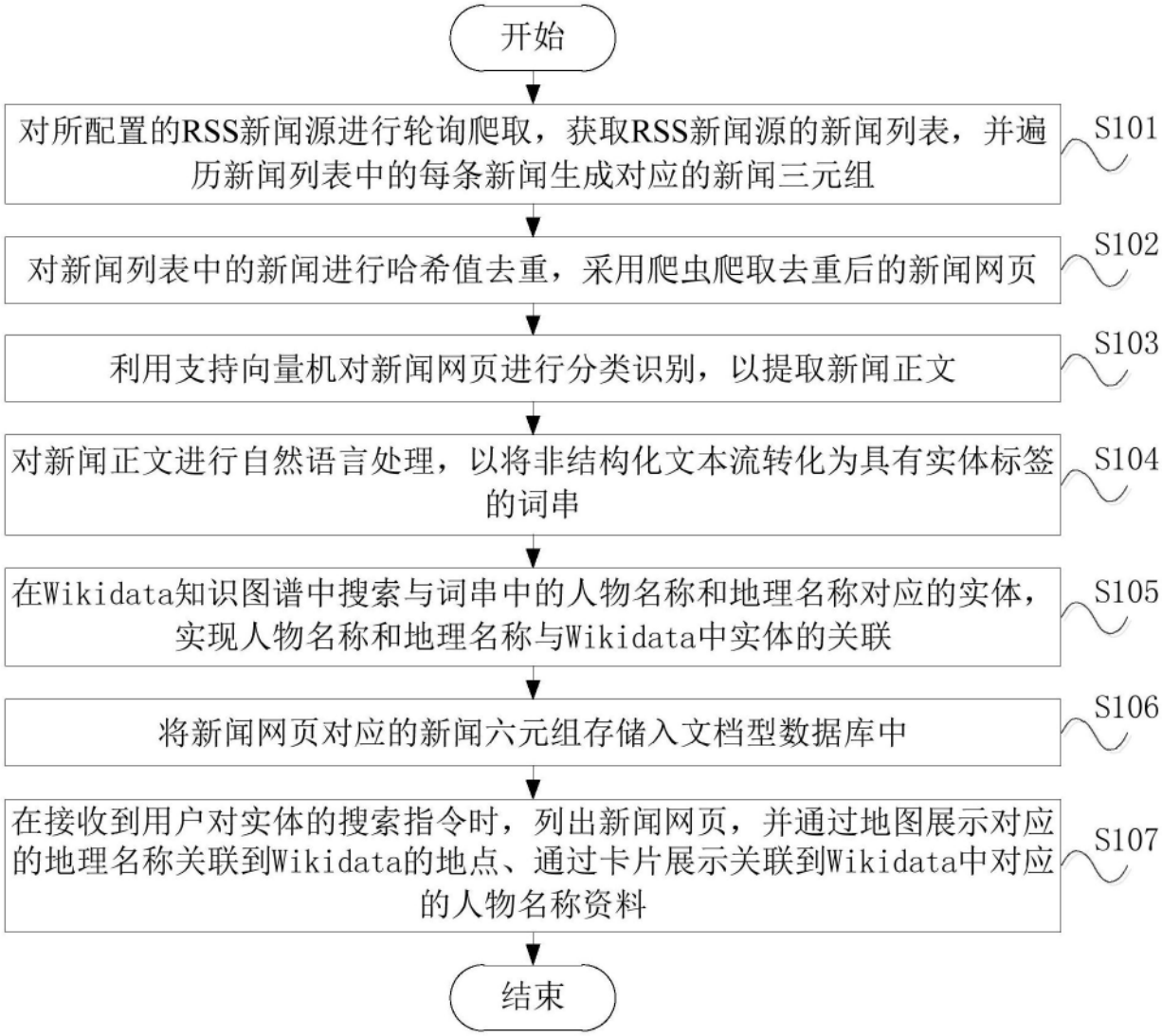

开始
对所配置的RSS新闻源进行轮询爬取，获取RSS新闻源的新闻列表，并遍历新闻列表中的每条新闻生成对应的新闻三元组
S101
对新闻列表中的新闻进行哈希值去重，采用爬虫爬取去重后的新闻网页
S102
利用支持向量机对新闻网页进行分类识别，以提取新闻正文
S103
对新闻正文进行自然语言处理，以将非结构化文本流转化为具有实体标签的词串
S104
在Wikidata知识图谱中搜索与词串中的人物名称和地理名称对应的实体，实现人物名称和地理名称与Wikidata中实体的关联
S105
将新闻网页对应的新闻六元组存储入文档型数据库中
S106
在接收到用户对实体的搜索指令时，列出新闻网页，并通过地图展示对应的地理名称关联到Wikidata的地点、通过卡片展示关联到Wikidata中对应的人物名称资料
S107
结束

图1

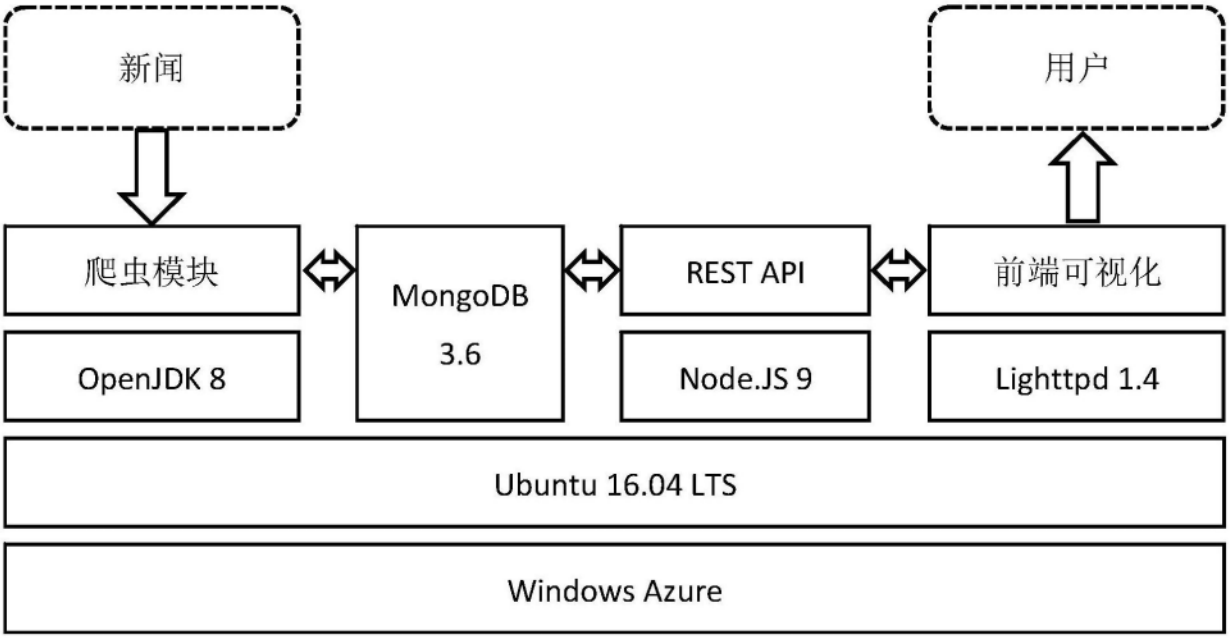

新闻
用户
爬虫模块
MongoDB
3.6
REST API
前端可视化
OpenJDK 8
Node.JS 9
Lighttpd 1.4
Ubuntu 16.04 LTS
Windows Azure

图2