

doi:10.3772/j.issn.2095-915x.2015.02.009

基于 OSGi 的大规模语义计算工具包平台^{*}

赖鸿昌, 朱礼军, 徐硕

(中国科学技术信息研究所信息技术支持中心 北京 100038)

摘要: 受大数据技术的影响, 科技情报分析工程化成为一种新的发展趋势。与论文和专利等事实型数据相类似, 适用于科技情报分析的工具软件必将成为一种新的重要资源。本文提出了基于 OSGi 框架大规模语义计算工具包平台。本文简述了相类似平台研究现状, 详细分析了大规模语义计算工具包平台的设计和实现, 汇报了当前工作进展, 通过具体案例详细介绍了平台的框架结构和功能, 同时指出了平台目前尚存在的问题, 并对平台的发展提出了展望, 期望借此可以打造科技情报分析工具的生态系统。

关键词: 语义计算, OSGi 架构, 资源共享

中图分类号: G350, TP311

Large-scale Semantic Computing Toolkits Platform based on OSGi

LAI Hongchang, ZHU Lijun, XU Shuo

(Center of Information Technical Support, Institute of Scientific and Technical Information of China, Beijing 100038, China)

Abstract: Influenced by the big data technology, the analysis of science and technology information engineering has become a new trend. Similar to the factual data such as papers and patents, the tools suitable for the analysis of science and technology information, will be a new important resource. This paper proposes the large scale semantic computing toolkits platform developed based on OSGi. In this paper, the content that is introduced firstly is the current situation of similar platform. Then, the design and implementation of large scale semantic computing toolkits platform are analyzed in detail, and the current work progress is reported. In the third section, the framework of platform is also introduced by a specific case in detail, including the potential problems in the development. Finally, the prospect of platform, to build an ecosystem of science and technology information analysis tool, is put forward.

Keywords: Semantic computing, OSGi framework, resources sharing

基金项目: 本研究得到国家自然科学基金项目“基于论文和专利资源的技术机会发现研究”(项目编号: 71403255)、“十二五”国家科技支撑计划课题“面向外文科技知识组织体系的大规模语义计算关键技术研究”(项目编号: 2011BAH10B04)、中国科学技术信息研究重点工作项目“大数据环境下融合多源信息的科技文献智能分析服务平台建设及应用示范”(编号: ZD2014-7-1)的资助。

作者简介: 赖鸿昌(1992-), 硕士研究生, 研究方向: 知识工程; 朱礼军(1973-), 博士, 副研究员, 研究方向: 知识组织、语义检索等; 徐硕(1979-), 博士, 副研究员, 研究方向: 数据挖掘、信息抽取等。通讯作者: 朱礼军, zhulj@istic.ac.cn。

1. 引言

在科研工作过程中, 经过长期的研究开发, 研究团队积累了一些系统工具, 也不可避免遇到诸如工具之间可重用性、集成性较差的问题。而这些工具软件作为一种科研共享资源, 并未受到足够的重视。在情报分析领域, 存在一些流程有较高的重用需求, 例如典型的文本相似度分析、文本语义分析等。而像分句、分词、词形还原等自然语言处理基本工具作为进行这些科研流程的基础, 也需要对相关的开源工具进行二次开发, 存在一定的开发难度。这并非是情报分析工作的核心任务, 却又占据了大量的时间, 一定程度上使得核心需求得不到满足, 增加了科研压力。将这些工具软件开源共享, 可以增进科研人员的交流, 降低开发成本和维护成本, 会更好地推进工具的技术创新和技术产出。

正是在这样的背景下, 我们研发了大规模语义计算工具包平台 SCT¹ (Semantic Computing Toolkits, 后文简称 SCT), 该平台可同时处理大规模非结构化资源 (如科技文献摘要) 和结构化语义资源 (如词表和概念等), 集成满足接口标准的词汇、概念层面的结构化计算、句子、篇章层面的语义角色标注、语义深层次标注等组件, 形成较为完整的面向大规模科技文献真实文本的语义计算工具包, 并作为工具包的共享平台。

文章的第二部分将介绍与 SCT 类似项目的概况, 第三部分介绍 SCT 平台的核心框架、应用实例及问题分析, 第四部分是总结和展望。

2. 大规模语义计算工具包平台现状分析

面对大数据时代所带来的海量数据危机, 为能让情报分析人员更专注于自己的核心业务, 很多项目为此进行了努力。Yahoo! Pipes 为解决信息过载提供了直觉的使用方式与弹性的架构, 为有需求的使用者提供了一个简单的混搭工具, 不受不熟悉的程序语法、数据类型、网络协议等技术门坎阻隔^[1], 但 Pipes 只能提供一套简单的 RSS 订阅功能, 并不能满足更深层次的需求。IBM 发布的 SystemML^[2] 将传统的机器学习算法转换为 MapReduce 的形式, 并提供了高级语言的程序在大规模集群编译和运行使用当中。微软于 2014 年发布了 Azure 机器学习实验室 (Machine Learning Studio)^[3], 基本理念与 Pipes 相类似, 利用拖拉功能模块的方式即可完成一个机器学习的流程, 但是缺乏自然语言处理的相关工具包。这些项目都为快速构建处理系统提供了很好的实践, 而 GATE 和 LarKC 则是文本处理的先行者。

2.1. GATE 概况

GATE (General Architecture for Text Engineering, 文本工程通用框架) 是英国谢菲尔德大学在 1995 年 1 月启动的项目, 曾是 EPSRC (Engineering and Physical Sciences Research Council) 项目在大规模信息抽取中的一部分^[4]。经过 20 年的发展, GATE 已经形成了文本处理完整生命周期的解决方案, 开发了如下产品:

表 1 GATE 的产品

IDE: GATE Developer	一个完整的语言处理开发环境, 打包了广泛使用的信息抽取系统的组件, 以及包括了词形还原、词性标注等插件在内的完整工具集。
Web 应用: GATE Teamware	工厂化语义注释项目的一个协助注释环境: 基于工作流引擎和一个重度优化的后端服务基础设施建立
框架: GATE Embedded	一个优化的对象库, 包含了针对不同应用的 GATE 开发者使用的所有服务的接口
架构	一个关于 GATE 的语言处理软件组成的高度组织化的图例
过程	关于服务的健壮性、鲁棒性和可维护性

1 平台地址: <http://168.160.18.215:8080/sct/>

GATE 之所以能够在 20 年里得到很好的延续，很大的原因是得益于其完整的核心框架使用了经典的 Java 组件模型，被拆解为可重用的块。GATE 的核心功能专注于工程最大的部分，比如特殊数据结构的建模和保留；标注、本体等的可视化和编辑；SVMLight、weka 等机器学习的插件实现等。GATE 的核心功能包括不同语言处理任务的组件，诸如信息检索、信息抽取等，代表性插件如 ANNIE (A Nearly-New Information Extraction System)。ANNIE 是 GATE 中典型的信息抽取系统，依赖于有限状态算法和 JAPE (Java Annotation Patterns Engine, Java 标注模式引擎) 语言。

GATE 在知识对象抽取中提供基础自然语言处理组件和灵活的开发接口，为进一步构建高效的知識抽取应用系统提供了充分的基础支撑^[5]。

采用了基于规则的信息抽取技术，在特定领域的自由文本抽取效果方面，比基于统计的信息抽取方法具有更好的效果。因此，将现有的资源和工具进行充分利用和共享，统一接口标准，集成常用工具，将是提高协同研究效率、解决一些工具性能瓶颈的有效途径。

2.2.LarKC 概况

LarKC(The Large Knowledge Collider, 大规模知识加速器)的目标是突破现在语义计算中存储、查询、推理等方面技术上的局限性。其基本设想是：这样一个基本架构必须要超越现在严格建立在逻辑上的典范。主要任务和创新包括^[6]：扩充现有基于逻辑的语义万维网推理方法；使用受认知科学启发的方法与技术；构建分布式推理平台。

LarKC 的总体架构如下：

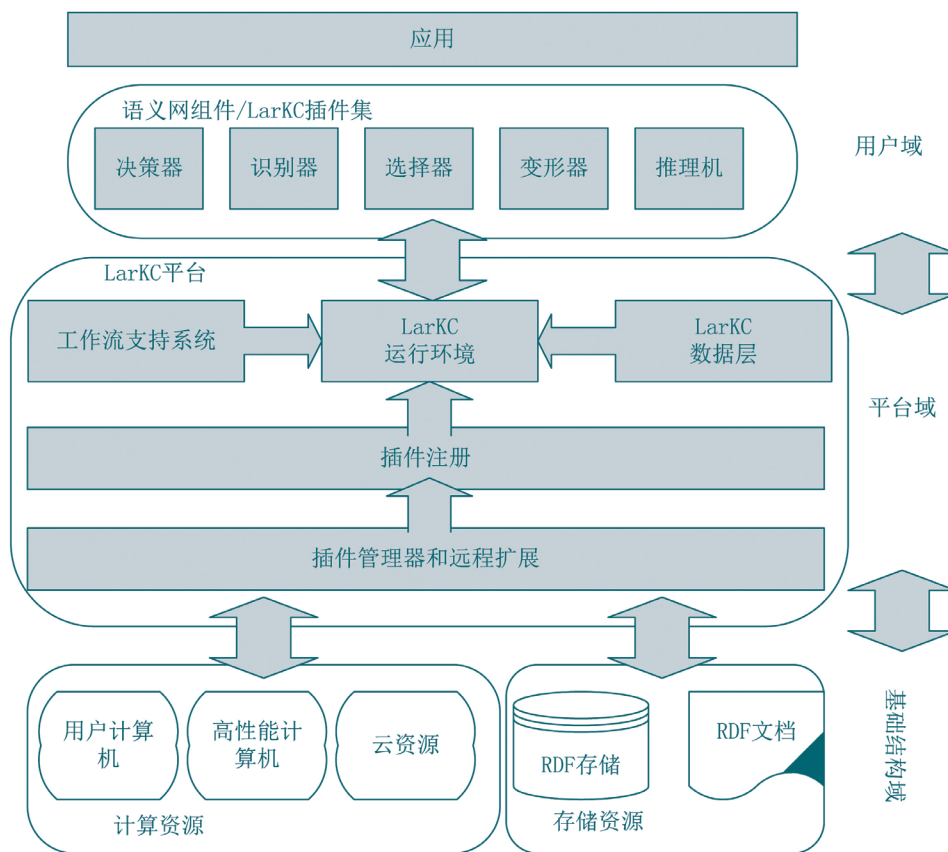


图 1 LarKC 架构总览^[7]

LarKC 架构有三个主要域（各种组件可以被组合起来）：用户域，包含应用、接口记忆语义网组件，以插件集的形式实现和提供；平台域，它连接了应用用户、语义服务以及多种硬件基础结构，包含了构成 LarKC 平台实际核心的特征；基础结构域，表征着各种类型的计算与存储资源，这可能对 LarKC 组件的部署与执行产生影响（包括来自用户域与平台域的组件）。目前，国内以 LarKC 项目为基础的研究成果主要有基于语义技术的智慧北京知识管理与决策支持系统^[8]、临床试验系统^[9]。

2.3. 小结

GATE 和 LarKC 项目作为文本通用处理框架和大规模工具集成的平台，在平台的架构设计上有很重要的借鉴意义。但是，GATE 和 LarKC 两个项目，所适用的客户更多是面对系统的设计者和开发者，在使用上对用户的门槛较高，具有一个相对陡峭的技术学习曲线，需要一段时间的学习才能应用至生产实践。从用户的角度来看，二者均只是提供了更完整的开发模板和系统方法，并未达到让用户“所见即所得”的应用目标。

大数据时代，情报分析工程化的趋势对 SCT 平台提出了更高要求：（1）快速响应：需要能对新需求建立快速的情报响应体系，要求能够基于已有的功能模块组件，通过流程的定制快速搭建数据处理和情报分析工具体系（2）动态更新：在流程任务的运行过程中，如果某个组件出现异常，系统需要自动将出现故障组件替换为可替代组件或者故障组件的另一个版本，使计算任务能顺利完成。（3）平台目标：设计一个可以实现最大化模块复用、易于扩展、符合“高内聚低耦合”标准的应用平台，并利用平台提供通用的组件接口，管理组件的注册、启动、停止和删除，更加强调接口而不是具体的实现。

3 基于 OSGi 架构的大规模语义计算工具包平台模型

OSGi(Open Service Gateway Initiative, 开放服务网关协议)技术是面向 Java 的动态模型系统，提供在多种网络设备上无需重启的动态改变构造的功能，最初是由 OSGi 联盟在 1999 年 3 月发起，主要目的是成为向网络设备传输服务管理的开放规范。OSGi 规范核心思想是一旦向网络设备中添加了一个 OSGi 服务平台，便可以在网络中的任意位置管理该设备上的服务组件。模块化(modularity)是 OSGi 规范的核心，集中体现为 Bundle 这一概念。OSGi 提供了一种面向服务的组件开发框架，具有高度模块化和动态化特点。而 OSGi 正好能用来解决如何更好的抽象服务，拆分复杂业务逻辑等诸如此类的问题。SCT 平台将 OSGi 架构引入，并对其做出了相应的评估。

SCT 平台在 OSGi 框架基础上，设计和开发稳定、健壮的语义计算工具包系统内核，支持组件的动态热拔插，而且单个或几个组件的崩溃不会影响整个系统的运行，支持增量式系统开发和持续集成，以便满足不断变化的用户需求。SCT 平台需要构建成有良好用户体验的服务平台，就必须做到对功能模块快捷高效的管理，既方便开发者进行迭代和开发，也使得用户能够更好得体验平台的功能。

3.1. OSGi 核心框架

OSGi 是 OSGi Alliance 组织制定的一个基于 Java 语言的服务（业务）规范一组规范，定义了 Java 的动态组件系统，由很多可重用的组件动态组成^[10]。OSGi 的生命周期层为 bundle 提供执行时模块管理和对底层 OSGi 框架的访问，使得能够从外部管理应用或者建立能够自我管理的应用（或者两者的结合），并且给了应用本身很大的动态性。OSGi 通过提供模块化的框架，大大降低了软件的复杂度，也为大规模分布式系统提供

了小型嵌入式应用程序。目前，OSGi 的框架已经大量应用到各类平台和应用框架^[11]中，在电力、军事等信息系统中应用广泛，在物联网中也逐渐配合传感器得到诸如设计家庭网关^{[[12]}杨旺明，徐守志. 基于 OSGi 的家庭网关系统的研究与设计^[12]等广泛应用。

从开发的角度来说，OSGi 具有以下特点：复杂性的降低；复用；简单；小巧；非侵入式。从部署和运行的角度来说，OSGi 的动态化特点包括：切合真实运行环境；易于部署；动态更新；适配性；透明；版本化；快速；懒加载^[13]。

本文通过对已有的标准体系结构框架进行综合分析，选择了 Eclipse 中的 Equinox OSGi 作为科技领域大规模语义计算的体系结构框架，借用标准体系结构的框架思想设计语义计算支撑工具包的系统基础框架。该框架允许用户根据需要轻松插入定制的分析组件，并将它们与其他组件进行组合。

3.2. 大规模语义计算工具包平台架构

OSGi 框架作为一个微内核的框架，由系统 bundle 管理所有 bundle 的生命周期，管理由各个 bundle 提供服务，并提供元数据管理、代码装载等功能。OSGi 框架中 bundle 可以在运行时通过服务注册中心选择一个可用的实现，bundle 可以注册新服务、接收关于服务状态的通知或者查找服务区以适配当前的设备。这使得一个 bundle 在部署后仍然具有可扩展性，新的 bundle 可被安装，已存在的 bundle 可修改和更新，而无需重新启动系统。

通过对 OSGi 规范的分析，可以得出 OSGi 对增量式开发的支持主要是基于：(1) 不断开发新 bundle 来添加新功能，从而满足新需求；(2) 通过代码调用或者服务调用来重用已开发 bundle 的功能^[14]。平台的基本架构如图 2 所示：

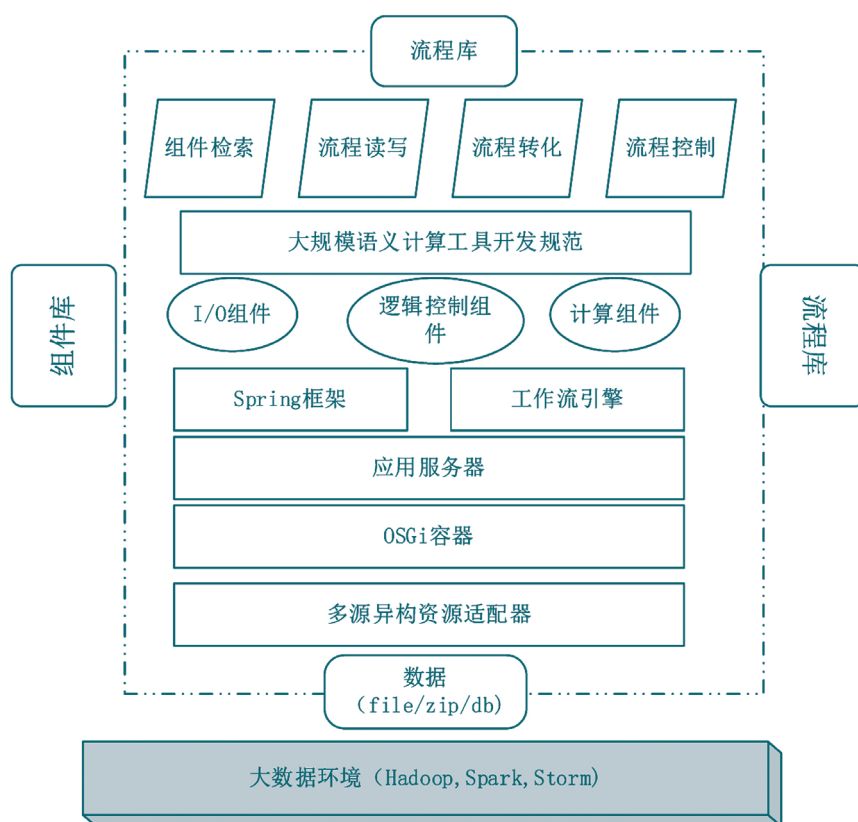


图 2 大规模语义计算工具包平台架构

平台基于 Spring 框架, I/O 组件、逻辑控制组件和计算组件作为平台的微内核, 构成平台的监管机制, 工具开发规范保证了各模块服务能够有良好的接口进行交互。多源异构资源适配器可以将各类资源转换成平台所能处理的数据格式。平台利用工作流 (workflow) 技术, 对工作流程及其各操作步骤之间业务规则的抽象、概括描述。利用工作流, 可以形成以下几个优点: (1) 改进和优化业务流程, 提高业务工作效率; (2) 实现更好的业务过程控制, 提高服务质量; (3) 提高业务流程的柔性等。在 GATE 和 LarKC 项目管理中, 流程的管理和多个服务或组件间包括信息、文档、任务的交互, 都离不开工作流技术。工作流引擎作为流程实例提供运行环境并解释执行流程实例的软件部件, 是平台保持良好运行状态的核心部分, 主要完成以下功能: 解释过程定义; 创建过程实例并控制其执行; 调度各项活动; 通过应用程序接口 (API) 调用应用程序; 提供监督

和管理功能等。

平台提供了组件库和流程库, 其中流程库是由组件构成的一个个完整的工作流程; 相当于同时提供了粗细粒度的功能模块, 既能够灵活组合, 又可帮助用户便捷使用。

3.3. SCT 平台概况

目前 SCT 平台主要包括 4 个模块: 流程空间、数据空间、组件空间、用户空间。

流程空间主要是依靠现有的组件依照一个正确的流程构建一个情报分析功能流程。流程设置好之后可以执行流程并动态显示日志, 及时向用户提供反馈信息。在流程管理中, 提供了包括日志、结果在内的下载, 包括分步 (依据组件) 下载和总结果下载, 可以查看流程的缩略图, 可以依据流程名和描述进行流程的检索; 如果需要执行相同流程, 可以利用复制功能得到同样结构的流程。

SCT	流程空间	数据空间	组件空间	用户空间
	创建流程	数据分类	组件分类	添加用户
	流程管理	上传数据	上传组件	用户管理
		数据管理	组件管理	

图 3 大规模语义计算工具包平台模块



图 4 流程空间 流程统计

数据空间和组件空间主要是负责数据和组件的上传、分类和管理。会对用户的意愿，将数据和组件区分为公开和私有。在某组件上传的过程中，需要选择该组件的前驱和后继，以便在用户创建流程时能够得到更好的指导，获得更好的用户体验。

用户空间为管理员提供了良好的用户管理功能。以复选框的形式，让管理员可以灵活地为用户分配角色、功能、权限。

3.4. SCT 平台应用实例

OSGi 架构可以很好地处理一个多模块级联的研究流程。以探究论文和专利资源领域深层主

题关联为例，描述一个典型的应用场景。论文和专利资源被认为是基本的科研指标，而论文和专利之间也有很多相互的和独有的关系。该应用场景主要解决了论文资源和专利资源相互之间的主题关联问题。

此前的有关专利和论文相关性的研究主要集中在专利对非专利文献的引用，虽然有成熟的引文分析方法可以借用，但很多论文未被引用，其中所含的有价值的信息便损失了。该过程体现为三个步骤：（1）利用概率主题模型，发现论文和专利语料库中的主题；（2）计算主题中的相似度；（3）构建专利和论文之间的主题关联^[15]。

经过分析，可以抽象为如下流程：

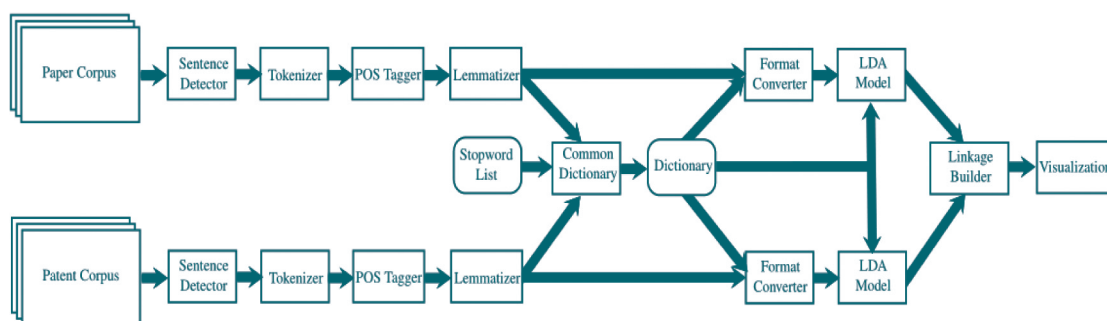


图 5 论文和专利深层主题关联分析流程

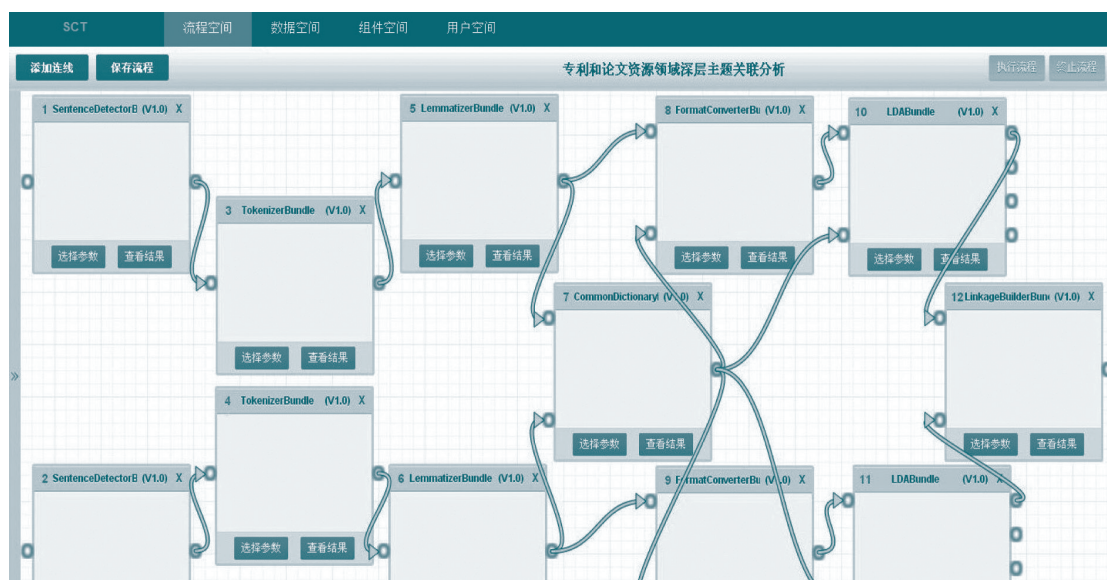


图 6 专利和论文资源领域深层主题关联分析

从图3中可以发现,在整个论文和专利深层主题关联分析流程中,论文和专利两个语料库都需要进行分句、分词、词性标注等基础性的工作,每一步都需要大量的I/O,而且代码量巨大。而语义工具包平台采用是面向服务的OSGi架构,基于OSGi的组件模型bundle能够隐藏内部实现,各个功能模块的bundle基于服务进行交互,只要处理好数据接口和语料库的格式,最后只需输入语料库,经过一定时间的计算,就能得出结果。在SCT平台中,将由负责各个任务的模块完成这项功能,如图6。

由于OSGi框架是动态的,bundle能够进行即时的更新,服务可以根据需要动态增加或者删除。OSGi帮助开发者处理了绝大多数动态性方面的工作。统一化的接口和标准定义了组件如何安装和管理,标准化的管理API使得OSGi能够与现有和将来的各种系统有机的集成。Bundle动态更新的特性使之能够动态的安装、启动、停止、更新和卸载,而整个系统无需重启,能很好地实现热插拔和动态集成。另外,平台提供了一套代

码原型Demo,减少开发人员的代码量,让开发人员将关注点集中于业务逻辑模型的设计;提供一套自动化的打包部署方案,简化应用程序的发布和部署。

在项目管理方面,bundle可以版本化,多版本能够共存而不会影响系统功能,解决了JAR hell^[16]的问题。在运行速度上,OSGi的类加载机制,和JAR包的线性加载不同,bundle委托式的类加载机制,使得类的加载无需进行搜索,且能有效的加快系统的启动速度。OSGi技术采用了很多懒加载机制。论文和专利深层主题关联分析流程中,由于懒加载机制,服务可以被注册,但是直到被使用时才创建,这使得运行时系统开销极小,而且能够灵活操作。

语义计算工具包是国内首个专门针对科技文献内容分析、处理工具的集成框架体系,通过开放的组件架构设计,可以为我国其他从事相关工作的各类研发群体提供一个可以复用的软件技术框架。目前有如下几类的若干个插件:

表2 现有组件的功能分类

基本功能类	分词,词形还原,词性标注,分句,格式转换
最小二乘支持向量机	包括最小二乘和多任务最小二乘向量机的选参、训练、预测的全部组件
语义相似度计算	基于双序列比对的中文术语语义相似度计算功能
依存句法分析	特征映射,模型训练
深层语义标注	浅层语义标注,深层语义标注

3.5. 存在的问题

模块化并不仅仅是把系统拆分成不同的块而已——这是JAR包就能做的事情,真正的模块化必须考虑到模块中类的导出、隐藏、依赖、版本管理、生命周期变化和模块间交互等一系列的问题。在SCT平台组件开发和使用过程中,可能会

遇到以下问题:(1)组件模块粒度控制问题,究竟多大粒度的组件更合适,目前尚无定论;(2)组件间上下游依赖关系问题,目前尚没有一个很好的机制自动完成上下游组件的推荐。

SCT平台真正的目的是打造一个开放的组件生态环境,做到共建共享,真正实现科研的共享、

交流的服务平台, 让更优秀的内容为同行所知和认可。但目前用户和组件数量仍然太少, 而且目前只支持 JAVA 编程语言, 而且不支持组件对数据库的操作, 这些很可能大大限制了平台的可扩展性。编程语言的限制可以通过 Web 服务的方式解决, 对数据库操作的支持将是未来重点需要考虑解决的问题。

4. 总结和展望

大数据时代, 情报分析工程化的趋势对基于云计算的平台提出了更高要求。大规模语义计算工具包平台提供了一个共享服务平台, 在快速构建科研工具的同时, 希望能够通过一种开放的模式构建一个科研共建共享的生态系统。GATE 和 LarKC 项目作为文本通用处理框架和大规模工具集成的平台, 在平台的架构设计上有很重要的借鉴意义。基于 Spring 框架和 OSGi 框架的 SCT 平台相对前两者而言对普通用户更加友好, 也更容易应用。另外, SCT 平台提供了组件库和流程库。其中, 流程库一般是用户提交的由组件构成的完整工作流程, 用于完成指定的任务。流程库和组件库的搭配组合, 相当于同时提供了粗细粒度的功能模块, 既能够灵活组合, 又可帮助用户便捷使用。另外, 由于引入了 OSGi 框架, 因此 SCT 平台具有诸如动态加载、易于部署、可扩展性好等特点。

大规模语义计算工具包平台作为一个共享服务平台, 在快速构建科研工具的同时, 希望能够以一种开放的模式, 构建一个科研共建共享的生态系统。但是目前用户数量和组件规模仍然是 SCT 平台的一个弱势, 而且对其他语言以及对数据库操作的支持性限制了平台的进一步发展。平台需要在其实体元素数目的可变性、结构关系的可调节性和结构形态的动态可配置性需要加强, 以满足 bundle 数量增加所带的模块管理难题。在

后续的版本中, 将会加入 weka 的默认数据格式 ARFF 作为默认的标准数据格式。另外, 对数据库操作的支持将是未来重点需要考虑解决的问题。

设计一个的软件框架, 需要考虑系统的健壮性, 扩展性, 安全性等多种因素。同时选择合适的工具和设计模式也是非常关键, 并且框架还需要在运行实践中不断完善。所以完成一个优秀的软件框架是一个复杂且长期的过程。

参考文献

- [1] 何银. Yahoo Pipes: 在数字化学习中实现精选 [J]. 中国信息技术教育, 2008(3): 66-67.
- [2] GHOTING A, KRISHNAMURTHY R, PEDNAULT E, et al. SystemML: Declarative machine learning on MapReduce[C]//2014 IEEE 30th International Conference on Data Engineering. Los Alamitos, CA, USA: IEEE Computer Society, 2011, 0: 231-242.
- [3] Introduction to Machine Learning Studio [EB/OL][2015-03-20].<https://msdn.microsoft.com/en-us/magazine/dn781358.aspx>.
- [4] GATE: a full-lifecycle open source solution for text processing [EB/OL][2015-03-20]<https://gate.ac.uk/overview.html>.
- [5] 刘建华, 张智雄. GATE 系统关键组件的应用开发接口研究和应用 [J]. 数字图书馆论坛, 2008(9): 26-31.
- [6] LarK 中文网站 [EB/OL][2015-11-15]. <http://www.wici-lab.org/wici/larkc/>.
- [7] Gallizo et al., D5.3.2 Overall LarKC architecture and design v1, September 2009.
- [8] 黄智生. 海量语义数据处理的最新技术和进展 [C]//中国计算机学会, 中国人工智能学会, 译. 第十二届中国 Rough 集与软计算学术会议、第六届中国 Web 智能学术研讨会及第六届中国粒计算学术研讨会联合学术会议论文集. 合肥: 2012: 4-4.
- [9] 黄智生. 基于语义技术的临床试验系统 [J]. China Digital Medicine, 2013, 8(4): 9-11.

[10] WhatIsOSGi [EB/OL][2015-03-20]<http://www.osgi.org/Technology/WhatIsOSGi>.

[11] 刘伟. 基于 OSGi 的家庭网关及其关键技术研究 [D]. 上海: 复旦大学, 2005.

[12] 杨旺明, 徐守志. 基于 OSGi 的家庭网关系统的研究与设计 [J]. COMPUTER AND DIGITAL ENGINEERING, 2010, 38(5): 159-162.

[13] 罗俊杰. OSGi 简介 [EB/OL] (2012-11-20)

[2015-05-14]. <http://osgi.com.cn/article/7289226>.

[14] 孙力军, 陈德人, 施敏华. 基于 OSGi 的自适应可进化软件框架 [J]. JOURNAL OF SOUTHERN YANGTZE UNIVERSITY(NATURAL SCIENCE EDITION), 2007, 6(2): 140-143.

[15] Xu S, Zhu L, Qiao X, et al. Topic linkages between papers and patents[C]//Proceedings of the 4th International Conference on Advanced Science and Technology, 2012: 176-183.