

SILK 关联发现框架综析^{*}

朱雯晶 夏翠娟 刘 炜

(上海图书馆上海科学技术情报研究所 上海 200031)

【摘要】对 SILK 关联发现框架进行深入分析,考察其不同的应用组件、关联发现引擎与分块技术等核心内容,并结合实例利用 SILK 工作台,介绍对两个数据源进行分析的方法以及关联发现的过程,探讨 SILK 框架解决 RDF 链接的关联问题的方法。最后总结 SILK 框架的特色、优势以及尚存的不足和有待解决的问题。

【关键词】关联数据 关联框架 关联发现 语义网

【分类号】TP391

Analysis of SILK Linkage Discovery Framework

Zhu Wenjing Xia Cuijuan Liu Wei

(Institute of Scientific & Technical Information of Shanghai, Shanghai Library, Shanghai 200031, China)

【Abstract】Based on SILK Link Discovery Framework, this paper analyzes its components, SILK Link Discovery Engine and Blocking Technology and so on. Using SILK Workbench, the process in finding RDF Links between two different data sources is introduced, and the methods which Silk Framework used in finding RDF Links are discussed. Finally, the merits and demerits of the SILK Link Discovery Framework are also analyzed.

【Keywords】Linked data Linkage framework Linkage discovery Semantic Web

1 引 言

对 Web 上已发布的数据集建立语义关联关系,是迅速普及关联数据应用、发现并重用已有数据的一个方便快捷的办法。目前,越来越多的网络资源正在向着数据网络(Web of Data,即 Web 中可被机器理解的语义数据)方向前进^[1],图书情报界如美国国会图书馆、OCLC,大型媒体公司如 BBC、纽约时报等都纷纷试水关联数据,大量传统网页上的数据正在被自动半自动地转换成关联数据,英国政府也已大规模提供语义网(关联数据)信息服务,Web 上的关联数据集越来越多,根据关联数据的四原则,数据一般使用 RDF 数据模型^[2],而数据之间的关联关系一般用“RDF 链接”来表示^[3,4],数据集中 RDF 链接的丰富程度是数据集质量的评价标准之一,RDF 链接越多,关联数据集的价值越高^[5]。不同发布方发布的关联数据集中有很多可以建立 RDF 链接的三元组,由于发布时无法统一考量而未实现关联,一些可以自动发现数据之间关联关系并生成 RDF 链接的关联发现工具应运而生,SILK^[6]就是一种关联发现框架,由柏林自由大学的 Christian Bizer 及其研究小组的两位助手与 Google 的工程师 Julius Volz 共同合作开发,目的是为了发现不同来源的关联数据集之间的关联关系,并同时让数据发布者在自己的数据和 Web 上的数据之间生成 RDF 链接。通过 SILK 框架,来自不同关联数据集中 RDF 数据之间的关联关系可以被机器自动发现并以机器可理解的方式将这些关联关系表达出来,在原本相互割裂的数据集之间建立起相互关联的桥梁,可以极大地丰富 Web 上数据集之间的语义联系,使数据产生更大的使用价值。

收稿日期:2013-03-14

收修改稿日期:2013-04-07

* 本文系国家自然科学基金重大项目“基于语义的馆藏资源深度聚合与可视化展示研究”(项目编号:11&ZD152)和国家自然科学基金项目“关联数据的理论和应用研究”(项目编号:11BTQ041)的研究成果之一。

2 关联发现框架的比较

包括 SILK 框架在内,用于发现和生成 RDF 链接的其他关联发现工具包括 LinQuer 与 GANT 等^[7]。与 SILK 相比较,这两者都只能按批次处理数据,而 SILK 有单机版和服务器版,更为灵活,可以随需求而变化^[8],SILK 的关联规则描述语言(Silk Link Specification Language, SILK - LSL) 可供开发人员自定义并配置对何种数据源进行关联、关联的条件以及需要进行的处理等。与 SILK 框架类似的还有 LIMES^[9] 和 R2R^[10] 框架,这两者各有所长,LIMES 框架在比较效率上较 SILK 更佳,但实现复杂;R2R 框架则需要熟悉 R2R 语言和编写 R2R 映射程序等,对于开发人员的要求相对较高。与之相比,SILK 框架支持大数据处理和服务端部署以及具有图形化界面的 SILK 工作台,因此拥有更加丰富的应用场景和潜力。三种重要关联发现框架的比较如表 1 所示:

表 1 三种重要关联发现框架的比较^[11]

| 框架 | 原理及架构 | 关联算法 | 功能特性 | 适用范围 |
|-------|--|---|---|--|
| SILK | 包括 SILK - LSL 语言规范和关联发现引擎,支持 SPARQL 查询多个关联数据集 | 基于字符串相似度算法、基于概念间的距离算法 | GUI 图形化用户界面,有单机版和服务器版,主要利用 owl: sameAs 来建立两个实体之间的关系 | 通用的 RDF 链接发现及生成工具,适合各种领域的关联数据集,支持大数据环境 |
| R2R | 由 R2R 映射语言规范和 R2R Java API 两部分组成 | 通过词表间的属性和值的映射关系定义从源数据集到目标数据集之间的关联规则 | 提供可被应用调用的 API,利用已有的术语词表映射将 Web 上的数据转换到应用的宿主词汇表 | 适用于术语词表的关联数据集之间的关联关系发现 |
| LIMES | 由 LIMES 规范语言和 LIMES 控制器组成,控制器包括输入输出模块、SPARQL 查询模块、数据缓存模块、算法驱动模块等 | 运用三角不等式建立粗索引预匹配,通过预匹配得到的示范实例过滤掉与示范实例明显不相关的匹配源,基于字符和数值的多种相似度算法 | GUI 图形化用户界面,既有 Web 版也有单机版;注重效率和性能;有去重功能;支持机器自动学习和后向关联 | 适合于大规模数据集之间的关联关系发现 |

3 SILK 原理及工作机制

SILK 关联发现框架的核心是关联发现引擎,所有的 SILK 框架内的组件都依赖这个组件进行运作,用 SILK - LSL 语言写成的关联规则描述文件是关联发现

引擎工作的依据,它定义了发现数据间关系的规则,供关联发现引擎读取。

3.1 SILK - LSL 语言^[12]

SILK - LSL 是一种结构化的声明式语言,用于指定 SILK 在哪些数据集之间发现 RDF 链接,一个典型的 SILK - LSL 文件如下:

```
<?xml version = "1.0" encoding = "utf - 8" ? >
<Silk >
<Prefixes... / >
<DataSources... / >
[ <Blocking... / > ]
<Interlinks... / >
[ <Outputs... / > ]
</Silk >
```

其结构上是一个标准的 XML 文件,根元素为:

<Silk > </Silk >。子元素包括:

(1) Prefixes: 指定数据集的前缀,如 <Prefixes > <Prefix id = "rdf" namespace = "http://www.w3.org/1999/02/22 - rdf - syntax - ns#" / > </Prefixes >。

(2) DataSources: 数据源定义,指定需要接入获取的本地或者远程 SPARQL Endpoint。

(3) Blocking: 数据项分块,在一般的比较中,两个数据集如果各包含 m 和 n 个数据项,则需要进行的比较运算是 $m \times n$ 的数量级。而数据项分块以后,则可以大大降低这个数量级。如对于唱片的信息进行比较,一般比较的数量级是 $m \times n$,而如果按照唱片公司来分组,两个数据集分别分为 100 组: $\sum_{i=0}^{100} m_i$ 和 $\sum_{i=0}^{100} n_i$,则分块后运算数量级降低为: $\sum_{i=0}^{100} m_i \times n_i$ 。

(4) Interlinks: 关联规则定义,其主要结构包括:

- ① LinkType: 生成的 RDF Links 类型。
- ② SourceDataset、TargetDataset: 源数据集与目标数据集。
- ③ LinkageRule: 关联规则,分为输入、变形、比较、聚合 4 种类型。

(5) Filter: 用于过滤生成的链接,可以指定每一个数据项返回最高置信度的 N 个链接,如 <Filter threshold = "0.7" limit = "1" / >。

(6) Outputs: 指定输出 RDF Links 的格式和位置。

3.2 关联发现引擎(SILK Link Discovery Engine)^[13]

作为 SILK 框架的核心,SILK 关联发现引擎起着最重要的作用。其数据流图如图 1 所示,从数据源中获取数据,并对其进行分簇处理,进行比较和发现关

联,最后过滤结果并进行输出。

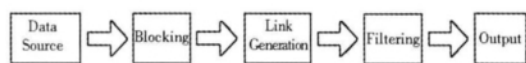


图1 SILK 关联发现引擎数据流^[14]

在这个过程中,核心的处理流程是 Blocking 与 Link Generation 两个阶段。Blocking 技术有效地降低了运算开销,大大提高了处理的性能和效果; Link Generation 则运用数据挖掘等相关技术,应用不同的相似度算法等,计算出两个数据项之间的差异程度,并以此作为计算后续阶段置信度的依据。

(1) 分块技术(Blocking)^[13]

在大数据时代,关联发现框架用于进行大数据集之间的关系发现,但是其中有很多的数据项比较是不必要的或明显不符合相关规则的,这部分比较将会浪费大量的计算资源。因此,许多降低这些浪费的方法被提出,最著名的就是分块技术。但是,分块技术通常都需要进行配置,且经常因为错误分块而导致的重新计算。

而在 SILK 框架中采用的是一种新的分块技术,这项技术是为不同的实体进行映射到一个多维索引表中去,如图2所示,通过经度纬度和名称标签,将不同的地点映射到一个三维索引表中去。映射到索引表中的每一个实体都保留了原本的距离,因此原本相关程度高的实体在索引表中的位置也会非常临近。

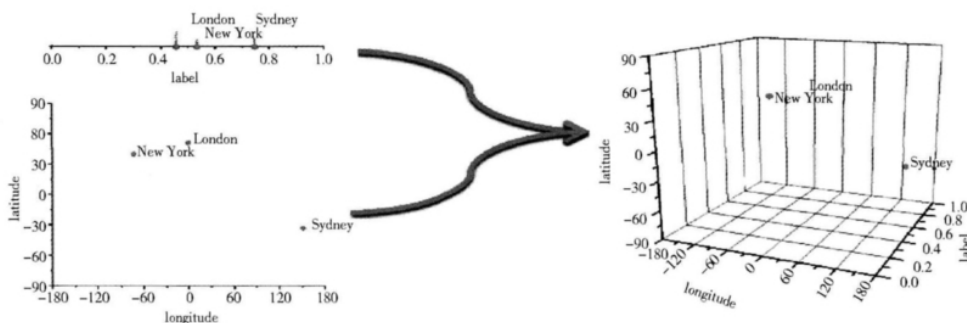


图3 二维与一维索引聚合^[16]

③比较对生成:通过前一阶段生成的索引,可以确定哪些实体对存在潜在的关联。随后这些生成的比较对将会通过关联规则进行评估计算,并得出最后的结果,以确定是否增加关联链接。

(2) 链接生成(Link Generation)

链接生成是 SILK 关联发现引擎最重要的阶段,通

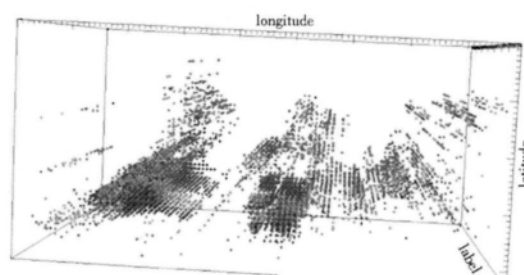


图2 SILK 分块技术多维空间索引^[15]

过数据挖掘等相似度计算,对于生成的不同的比较对进行计算,以确定最后的置信度。对于每一个比较对将会计算相似度,相似度的值区间可以认为是一个百分数^[17]。

SILK 框架中,分块技术分为三个阶段:

①索引生成:在每一个相似度算法中,都要通过一系列在欧几里得空间中定义的向量组创建索引,不同的索引方式将会决定每一个实体所需要的向量个数。同时,向量空间的维数则由数据类型决定,如纯数值域只需要一维向量来标识距离,而空间点坐标需要经纬度二维向量来标识距离。

②索引聚合:这个阶段中,所有在索引生成阶段生成的不同索引将会被整合进一个符合索引中去。这样不同的实体在保留原有的相对距离的同时又拥有相同的索引向量,如图3所示。将二维的距离向量索引与一维的字符串向量进行索引聚合,会生成一个三维的索引空间。

在这个过程中,最关键的是关联规则(Linkage Rule)^[18]。在 SILK 框架中,关联规则分为4部分:路径

输入,从一个实体的指定 RDF 路径中获取值,如?movie/rdfs:label; 变形 将一个数值类型进行转换处理,如 lowerCase; 比较 将两个输入的值以用户指定的方式进行距离计算,并返回置信度; 聚合 将不同算法得出的置信度进行聚合,返回最后的结果。

① 路径输入 (Path Input)

路径输入阶段指定了哪些实体的属性值将会被抽取并进行比较。在路径输入中,可以通过在数据集中已经定义好的变量名加路径来获取不同的属性值,其中可以使用的操作符如表 2 所示:

表 2 SILK 关联规则路径输入操作符^[19]

| 操作符 | 名称 | 用法 | XML 实例 |
|-----|-----|--|---|
| / | 前进符 | <路径> / <属性> | #获得影片的名称 < Input path = "? movie/rdfs: label" /> |
| \ | 返回符 | <路径> \ <属性> | #获得指定艺术家的所有唱片列表 < Input path = "? artist \ dbpedia: artist [rdf: type = dbpedia: Album]" /> |
| [] | 过滤符 | <路径> [<属性> <运算符> <值>] <路径> [<lang> <运算符> <值>] | # 获得影片的英文名称 < Input path = "? movie/rdfs: label [@ lang = 'en'" /> |

② 变形 (Transformation)

不同的数据集可能有不同的编码或其他格式,通过变形处理,可以在比较前将这些格式进行统一处理,以返回正确的比较结果格式。SILK 支持的变形处理提供了丰富的处理方式对数据进行清洗,如表 3 所示:

表 3 变形处理类型^[20]

| 名称 | 功能 |
|-----------------------------|------------------|
| Alpha Reduce | 去除所有非英文字母的字符 |
| Capitalize | 将单词首字母大写 |
| Concatenate/MultipleValues | 将两个输入的内容组合成一个 |
| Convert Charset | 编码转换 |
| Logarithm | 将输入的数字计算对数,可选择底数 |
| Lower Case/Upper Case | 转换全小写、全大写 |
| Merge | 将输入的内容合并 |
| Numeric Reduce | 将所有非数字的输入移除 |
| Regex Replace | 正则表达式替换 |
| Remove Blanks/empty values | 去除空格或者空值 |
| Remove Special Chars/Values | 去除特殊字符/特定字符 |
| Stem | 用波特词干算法处理 |
| Strip URI prefix/postfix | 去除"http://"/去除后缀 |
| Tokenize | 令牌化 标记解析 |

③ 比较 (Comparison)

比较阶段,两个经过处理后的输入将会通过用户定义的距离计算方法和阈值进行比较运行,以得到结果。输出的结果为 0 则为相同,超过 0 则为不完全相同,如图 4 所示,在 0 到阈值之间的结果将会产生一个正分,否则将会是负分。因

此,对于阈值的选择将会影响最后结果的准确程度。

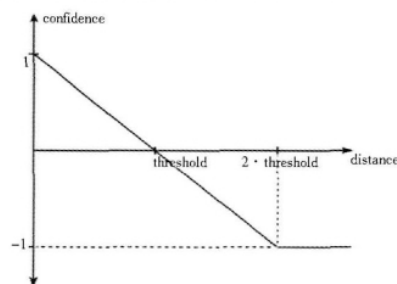


图 4 阈值与置信度、相似度的关系^[21]

比较阶段采用的相似度算法主要分为:

1) 基于字符的距离计算:从字符层面对两个字符串进行比较,通常对字母拼写错误等有较好的效果,SILK 支持的算法如表 4 所示:

表 4 基于字符的距离算法^[21]

| 名称 | 功能 |
|---------------------|---------------------------------------|
| levenshteinDistance | Levenshtein 距离算法 |
| levenshtein | Levenshtein 距离算法,将结果标准化处理至区间 [0, 1] 内 |
| jaro | Jaro 距离,多用于比较人名 |
| jaroWinkler | Jaro - Winkler 距离算法,用于人名等短字符串 |
| equality | 字符串相同返回 0,否则返回 1 |
| inequality | 字符串不相同返回 0,否则返回 1 |

2) 基于语句的距离计算如表 5 所示:

表 5 基于语句的距离算法^[21]

| 名称 | 功能 |
|-------------|----------------|
| jaccard | Jaccard 距离计算 |
| dice | 相似度计算 |
| softjaccard | 弱 Jaccard 距离计算 |

3) 其他用途的距离计算如表 6 所示:

表 6 其他用途的距离算法^[21]

| 名称 | 作用 |
|---------------------------------------|---|
| num (float minValue, float maxValue) | 计算两个数值之间的差距; 参数含义 minValue, maxValue: 在数据源中出现的最小和最大数值 |
| date | 计算两个形如“YYYY - MM - DD”日期差异,返回差异天数 |
| dateTime | 计算两个符合 xsd: dateTime format 的日期时间的差异,返回差异秒数 |
| wgs84(string unit) | 计算两点之间的地理距离; 参数含义 unit: 单位,如米“m”,千米“km” |

④ 聚合 (Aggregation)

聚合阶段需要将不同的多个置信值处理为一个单值。因为大多数情况下,确定两个实体是否相同,并不会简单比较一个属性值。在 SILK 中,聚合处理的类型如表 7 所示。有些算法可以支持加权,使得所得的置信结果更加精确和可靠。

表 7 聚合处理类型^[22]

| 名称 | 作用 |
|---------------|-------------------|
| average | 处理获得(加权)平均置信值 |
| max | 取最大的置信值 |
| min | 取最小的置信值 |
| quadraticMean | 用欧氏距离进行聚合 |
| geometricMean | 处理(加权)一组置信值的几何平均数 |

4 SILK 应用实例

通过使用 SILK 工作台可以非常方便地对各种数据源进行发现。本文以 SILK 工作台为工具,简述如何利用 SILK 工作台对 DBpedia 的 Settlement 与 LinkedGeoData 中的 City 之间进行关系发现。

首先启动 SILK Workbench 并用浏览器访问首页,新建项目,并添加项目中需要的前缀,如 dbpedia: <http://dbpedia.org/ontology/> dc: <http://purl.org/dc/terms/>等。

然后配置数据源,分别配置 DBpedia 与 LinkedGeoData 的数据源,还可以配置并修改一些常用的参数,包括页大小、重试计数、重试间隔等。

随后添加关联任务,配置源和目标数据源,如图 5 所示,分别选择之前已经配置好的前缀,并配置相关的限制语句,在链接类型中,默认为 owl: SameAs 类型。

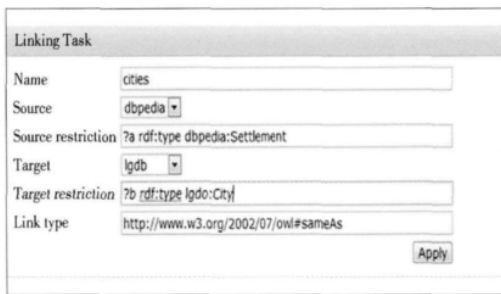


图 5 SILK 工作台配置关联任务对话框

在添加好关联任务后,可以打开编辑关联规则, SILK Workbench 提供了可视化的关联规则编辑器,可以让用户方便地编辑关联规则,如图 6 所示:



图 6 SILK 工作台关联任务编辑器主界面

左边栏分别显示了 3.2 节中的 4 类组件,包括当前源和目标数据源实体的各属性值与变形、比较、聚合组件。右侧窗口为编辑窗口,通过鼠标点选,将左侧的内容拖放到右侧区域中,不同的组件用不同的颜色进行标识。

此关联任务完整的关联规则分别将两个数据源中 Label 的英文名称与默认名称进行 Levenshtein 距离计算,随后通过聚合组件取两者的最大值进行处理。同时,通过 wgs84 地理位置距离计算,将最后的结果进行平均值聚合处理,以得到最后的置信值,如图 7 所示:

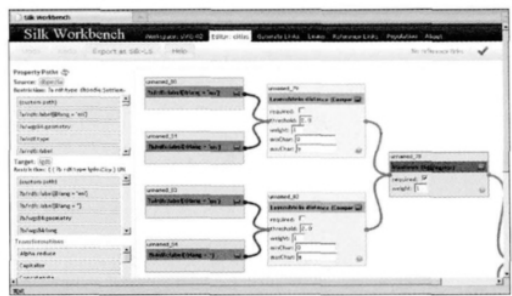


图 7 关联规则编辑器实例

当配置完成以后,可以运行关联任务,查看最终的结果,如图 8 所示:



图 8 SILK 工作台关联任务执行结果评估页

除了可以实时查看已经发现的关联,还可以查看每一步关联规则的结果、置信度。在图 8 中,迪拜的名称匹配度为 100% 完全相同,而地理位置信息计算结果为 99.2% 的置信度,最后平均得到 99.6% 的置信度。可以认为,在 DBpedia 中的“Dubai”可以与 LinkedGeoData 中的“node31248510”添加 owl: SameAs 链接。

最后,如果对于关联任务的执行结果没有疑问,且关联规则配置无需进行再次修改,则可以再次执行关联任务,并选取之前配置好的输出任务,将结果输出到

指定位置。输出的形式同样可以进行配置,本例中输出格式为 N3,N3 文件的默认输出位置为操作系统当前用户主目录的 SILK 目录下的 output 文件夹。

5 结 语

SILK 框架的优点在于通过提供不同的组件应用于不同的场景,包括单机与大数据环境均可使用,且 SILK 提供自带的 SILK 工作台方便用户使用并简化了操作步骤与复杂度,包括自动生成 SILK-LSL 文件;通过图形化的关联规则编辑器进行编辑;可视化的结果评估窗口,使用户可以快速调整和改进关联规则;提供了 REST API^[23] 供用户集成并调用,进一步拓展了 SILK 的应用环境。同时,SILK 引擎中所采用的分块技术等也极大地提高了效率。SILK 官方列出了 SILK 单机版和大数据版利用分块技术和不利用分块技术运行的对比表。在 DBpedia 与 LinkedGeoData 中建立 owl:sameAs 链接时,利用了两个数据源的子集,分别包含 105 000 与 59 000 条数据,测试环境分别为 SILK 单机版: Intel Core2Duo E8500 处理器、8GB 内存; SILK 大数据版: 亚马逊 Elastic MapReduce Cluster,包含 10 个亚马逊的云计算机(Amazon EC2)。在没有应用分块技术时,总的执行比较次数超过了 60 亿次;而应用分块技术以后,分块技术将城市按照名称划分为 50 个分簇,随后得到的结果如表 8 所示:

表 8 SILK 单机版与大数据版性能对比

| | 版本 | 链接生成时间 | 链接生成数 |
|-------|-----------|----------|-------|
| 无分块技术 | SILK 单机版 | 54 小时 | 9 283 |
| | SILK 大数据版 | 6.7 小时 | 9 283 |
| 有分块技术 | SILK 单机版 | 2.592 小时 | 9 224 |
| | SILK 大数据版 | 0.24 小时 | 9 224 |

从表 8 可以发现,利用分块技术与大数据版可以大大降低运行的时间,提高效率,且生成的链接数丢失率相比无分块技术不超过 1%。

而 SILK 框架也有一些不足之处,它对计算资源的要求较高,在一些普通配置的计算机上可能会发生启动时间较长、部分操作响应较慢的问题。同时,SILK 在处理数据时,需要首先将数据源的内容缓存到本地,在缓存的过程中,受网络环境、数据源影响较大,且数据缓存无法主动释放,可能会导致 Java 虚拟机内存溢出。还有包括对国内的部分主流浏览器如 IE 浏览器的兼容性不足,SILK 内置的相似度算法多不支持汉字

字符等问题。

相信在关联数据、语义网技术不断发展的过程中,SILK 框架通过与大数据、数据挖掘等相关领域的技术进行结合,会在用户研究应用的过程中提供更好的技术支持与保障。

参考文献:

- [1] Bizer C, Heath T, Berners-Lee T. Linked Data - The Story So Far[J]. *International Journal on Semantic Web and Information Systems*, 2009, 5(3): 1-22.
- [2] 刘炜. 关联数据: 概念、技术及应用展望[J]. *大学图书馆学报*, 2011, 29(2): 5-12. (Liu Wei. Overview on Linked Data: Concept, Technology and Implementation[J]. *Journal of Academic Libraries*, 2011, 29(2): 5-12.)
- [3] Bizer C, Cyganiak R, Heath T. How to Publish Linked Data on the Web[OL]. [2011-05-10]. <http://wifo5-03.informatik.uni-mannheim.de/bizer/pub/LinkedDataTutorial/>.
- [4] 夏翠娟, 刘炜, 赵亮, 等. 关联数据的发布技术及其实现——以 Drupal 为例[J]. *中国图书馆学报*, 2012, 38(1): 49-57. (Xia Cuijuan, Liu Wei, Zhao Liang, et al. The Current Technologies and Tools for Linked Data: A Case of Drupal[J]. *Journal of Library Science in China*, 2012, 38(1): 49-57.)
- [5] Heath T, Bizer C. Linked Data: Evolving the Web into a Global Data Space[M]. Morgan & Claypool Publishers, 2011: 1-136.
- [6] Isele R, Jentzsch A, Bizer C, et al. Silk - A Link Discovery Framework of the Web of Data[EB/OL]. [2012-07-30]. <http://www4.wiwiwiss.fu-berlin.de/bizer/silk/>.
- [7] Hassanzadeh O, Xin R, Miller R J, et al. Linkage Query Writer [C/OL]. In: *Proceedings of the 35th International Conference on Very Large Data Bases*. 2009. [2011-11-22]. <http://dl.acm.org/citation.cfm?id=1687599>.
- [8] Isele R, Jentzsch A, Bizer C. Silk Server - Adding Missing Links While Consuming Linked Data[OL]. [2012-07-30]. <http://www4.wiwiwiss.fu-berlin.de/bizer/silk/>.
- [9] LIMES - Link Discovery Framework for Metric Spaces[EB/OL]. [2011-11-22]. <http://aksw.org/Projects/limes>.
- [10] The R2R Framework - Transforming RDF Datasets, User Manual and Mapping Language Specification[EB/OL]. [2011-11-22]. <http://www4.wiwiwiss.fu-berlin.de/bizer/r2r/spec/>.
- [11] 陶俊, 孙坦. 基于 Linked Data 的 RDF 关联框架综析[J]. *现代图书情报技术*, 2011(12): 1-8. (Tao Jun, Sun Tan. Analysis of Framework for RDF Linkage Based on Linked Data[J]. *New Technology of Library and Information Service*, 2011(12): 1-8.)
- [12] Link Specification Language[EB/OL]. [2012-06-18]. <http://www4.wiwiwiss.fu-berlin.de/bizer/silk/>.

- tps://www.assembla.com/spaces/Silk/wiki/Link_Specification_Language.
- [13] Silk Link Discovery Engine [EB/OL]. [2011-10-12]. https://www.assembla.com/spaces/Silk/wiki/Silk_Link_Discovery_Engine.
- [14] Typical Workflow of Creating a New Link Specification [OL]. [2012-06-13]. https://www.assembla.com/spaces/Silk/documents/bvi67yA10r4iv_eJe5cbLr/download?filename=workbench_workflow.png.
- [15] Index Generation, Index Aggregation and Comparison Pair Generation [OL]. [2012-06-13]. <https://www.assembla.com/spaces/silk/documents/aGd8h8x5Kr4iPEeJe5cbLr/download?filename=plotLabeled.png>.
- [16] Indexing [OL]. [2012-06-13]. https://www.assembla.com/spaces/silk/documents/b3_7o8x5Or4iyIeJe5cbCb/download?filename=indexing.png.
- [17] Silk Workbench [EB/OL]. [2012-06-13]. https://www.assembla.com/spaces/silk/wiki/Silk_Workbench.
- [18] Linkage Rule [EB/OL]. [2011-10-12]. https://www.assembla.com/spaces/silk/wiki/Linkage_Rule.
- [19] Path Input [EB/OL]. [2011-09-30]. https://www.assembla.com/spaces/Silk/wiki/Path_Input.
- [20] Transformation [EB/OL]. [2011-11-07]. <https://www.assembla.com/spaces/Silk/wiki/Transformation>.
- [21] Comparison [EB/OL]. [2011-12-15]. <https://www.assembla.com/spaces/silk/wiki/Comparison>.
- [22] Aggregation [EB/OL]. [2011-11-06]. <https://www.assembla.com/spaces/silk/wiki/Aggregation>.
- [23] Silk Core 2.5.2 API [EB/OL]. [2012-03-06]. <http://wifo5-03.informatik.uni-mannheim.de/bizer/silk/api/current/#package>.
- (作者 E-mail: wjzhu@libnet.sh.cn)

《现代图书情报技术》特邀专栏组稿

《现代图书情报技术》是中国科学院主管、中国科学院国家科学图书馆主办的计算机信息管理技术方面的学术性刊物。刊物拥有清晰的定位,即以跟踪技术的研究、应用、交流为主体,服务于广大信息技术人员。

本刊从 2004 年起开设不定期栏目——《特邀专栏》,每期专栏集中发表关于某个特定方面的技术研发与应用的研究型文章,汇集科研成果、聚焦研究前沿。

1 《特邀专栏》操作办法及流程

(1) 本栏目特邀国内外知名专家、学者、教授担任专栏主编,专栏的设立一般由期刊的策划编辑和特邀专栏主编沟通,根据国内外图书情报技术学科的发展需要提出选题。

(2) 选题一旦确定后,由特邀专栏主编承担稿件的组织,审核并撰写前言。一期特邀专栏一般为 4-6 篇文章为宜。稿件组织过程中,策划编辑将与特邀专栏主编进行定期的沟通,及时掌握稿件的撰写情况,并对稿件的撰写提出适当的建议和意见。

(3) 稿件经特邀专栏主编审核通过,提交给编辑部。后期由策划编辑负责与作者的联系沟通及安排出版等事宜。

(4) 专栏的选题一旦确定后,将确定基本时间表。一般的操作周期为 3-5 个月。以正式确定特邀专栏题目为起始点,在 1 个月内确定约请论文的作者和题目,3 个月内确定初稿,5 个月内确定采用稿。

2 《特邀专栏》稿件内容要求

(1) 深入反映本专栏选题方向的前沿研究成果或重大应用成果,侧重理论研究、技术分析、系统论证或设计等,注意理论与实践相结合。

(2) 特邀专栏稿件应该主要是原始性和原创性研究论文,也可以有一篇综述性论文,但综述性论文必须可靠地覆盖该方向的原始核心文献。

(3) 文章按照严谨的学术文章体例写作,即明确扼要地界定研究问题,简要说明研究方法,系统精炼地描述国际国内发展状况,进而详细地描述作者自身研究工作的技术线路及研究结果。

(4) 特邀专栏的一系列文章应注意覆盖专栏选题所涉及的各个研究方向和多个研究单位,充分覆盖可能存在的多种观点和技术线路。

(5) 充分承认前人/别人的工作,充分引证所参考引用的文献(尤其是本研究工作中的原始核心文献和国内最先出现的研究文献),严格遵守著录规范。

3 《特邀专栏》稿件格式要求

(1) 论文版式请参照本刊网站“下载专区”中“论文模板”。

(2) 多个作者时,请注明通信作者,并注明各个作者的单位。

(3) 每篇稿件以 6-8 千字为宜(按篇幅字数计算,包括图、表)。