Falcons Explorer：一个面向语义 Web 的实体探索系统

吴辉耀[1]，胡伟[2]
（1. 东南大学计算机科学与工程学院，南京 210096；
2. 南京大学计算机软件新技术国家重点实验室，南京 210093）

摘要：随着越来越多语义 Web 数据变得可用，本文提出了一种结合实体搜索和表格式编程的方法来帮助用户更好探索语义数据和更准确地表达信息需求。基于该方法构建的在线系统 Falcons Explorer 封装了语义 Web 搜索引擎 Falcons 提供的搜索服务和数据检索 API，提供基于关键字的实体搜索。搜索结果使用传统列表和表格两种视图进行呈现。特别地，结果视图支持表格式编程操作，隐式地构造结构化查询，增强了用户探索数据的能力。

关键词：语义 Web；浏览；实体搜索；表格；终端用户编程
中图分类号：TP312

Falcons Explorer: An Entity Exploration System for Semantic Web

WU Huiyao[1], HU Wei[2]
(1. School of Computer Science and Engineering, Southeast University, Nanjing 210096;
2. State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093)

Abstract: With more and more Semantic Web data available for the public, to help end users specify information need and explore Semantic Web data better, in this paper we propose a solution that combines entity search with tabular-based programming. Based on the proposed solution, we develop a system named Falcons Explorer, which is grounded on the search service and an API for data retrieval provided by Falcons. It utilizes the keyword-based method to start entity search, and provides a tabular-based view to show search result in addition to list view. In particular, tabular-based view can help user expore semantic data better and increase the query expressive power with the tabular programming.

Key words: Semantic Web; Browsing; Entity Search; Table; End-User Programming

0 引言

随着语义 Web 的不断发展，整个语义 Web 已经产生了大量的语义数据。W3C 的社区项目 Linking Open Data 已经汇集了数十亿条的 RDF 三元组[1]，从 2007 年只有 12 个数据集到 2011 年的 295 个数据集，覆盖了地理位置、生物医学、出版物和电影等众多领域。可以预见，不远的将来整个语义 Web 将变成一个巨大的数据之网。

帮助终端用户高效和方便地消费如此大量的语义数据是当前语义 Web 发展过程中一个比较迫切的任务，传统基于关键字的搜索是一个比较有效的方法，比如语义 Web 搜索引擎 Falcons[2]。然而，基于关键字缺少足够的语义信息很难准确理解用户的查询需求，在文献[3]提出了一种增量查询构建的方法，支持交互式地引导用户确认查询的意图。但有时候，用户一开始并没有比较明确的查询需求，而目前传统搜

基金项目：高等学校博士学科点专项科研基金（课题编号: 20100091120041）
作者简介：吴辉耀（1986-），男，硕士研究生，主要研究方向：语义 Web
通信联系人：胡伟（1982-），男，博士，讲师，主要研究方向：语义 Web，本体工程，数据融合. E-mail: whu@nju.edu.cn

索引擎返回的搜索结果列表并不支持用户做进一步的探索，要求用户遍历每条搜索结果记录是一种比较低效的方式。

本文提出了一种方法，结合传统的实体搜索和表格式编程方法，帮助用户简单直观地完成复杂查询的表达。表格式编程是一种支持用户在表格视图上完成一些查询操作的过程，比如对列的添加和删除，根据列的值对行进行过滤等。由于表格广泛存在于电子表格和数据库中，是面向普通用户组织信息的一种流行的结构，据报告有大约 60%的终端用户在使用电子表格或者数据库，且有将近一半使用电子表格的用户都会使用条件陈述（比如 IF）或者公式[4]。换言之，普通用户除了在日常工作中使用电子表格外，还会在表格上做一些编程操作。因此，基于表格呈现信息的方式更能让大多数用户接受，且可以有效地降低学习的成本。

该方法具有如下几个主要特点：1）基于关键字的实体搜索作为入口，增加系统的可用性，减少用户学习的代价；2）提供实体类型的层次结构，可以实现快速过滤用户感兴趣的实体；3）基于表格的实体搜索结果信息呈现，提供比现有实体搜索引擎更强的查询表达能力，同时有利于用户对实体信息的探索。

本文第 1 节介绍了基于本文提出的方法所开发的系统 Falcons Explorer 的架构，第 2 节描述了基于 Falcons 的实体搜索的界面和功能介绍，第 3 节详述了基于表格视图的编程，最后是对本文提出方法的总结和未来工作的展望。

1 系统的架构

基于本文提出的方法开发了 Falcons Explorer[1]在线系统，该系统服务端基于 Java 技术实现，客户端主要采用了 jQuery 框架和 Ajax 技术。

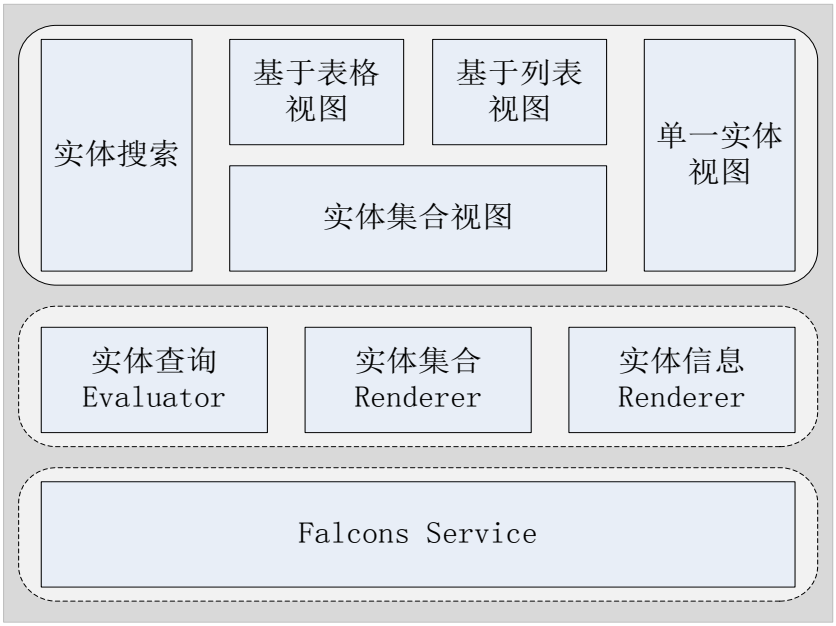


图 1 系统架构

Fig.1 System architecture

[1] http://ws.nju.edu.cn/explorer/

该系统的架构如图 1 所示，整个架构基于 MVC 设计，主要包含三个部分：视图层（最上层）、逻辑控制层（中间层）、数据模型层（最下层）。

在视图层，提供了实体搜索功能，以及相应单一实体和实体集合视图，对于实体集合提供了两种主要视图：1）基于表格视图，提供了一些表格操作的功能，支持面向终端用户的编程；2）基于列表的视图，提供了实体基本信息预览，并提供了实体的类型层次结构，支持基于类型的过滤和导航。

在逻辑控制层，根据用户输入的关键字（实体搜索）、访问的实体和针对实体集合所构建的查询来控制实体（集合）视图信息的呈现。

数据模型层，主要基于语义 Web 搜索引擎 Falcons 提供的搜索服务和数据检索 API 来获取相应的数据，包括基于关键字查询匹配的实体集合，查询实体的信息，获取实体数据源，获取类型的层次结构等，这些数据被封装到相应的数据结构，传输给上层。

服务器和客户端主要基于 Ajax 技术进行通信，传输的数据格式为 XML。

2 基于 Falcons 实体搜索

语义 Web 搜索引擎 Falcons 提供了基于关键字搜索实体的能力，背后的数据来自于 Falcons 爬虫从真实的 Web 中爬取的语义数据。目前 Falcons 实体搜索引擎，依旧是按照传统搜索引擎的方式来组织搜索结果，即以列表方式呈现，每个匹配到的实体都会提供一段 snippet 作为命中的证据。由于关键字缺少足够的语义表达能力，因此，用户往往需要对返回搜索结果做进一步限制，比如搜索所有名字包含 Tim 的人，为了进一步了解这些返回实体的信息比如工作单位以限制用户所要查找的实体，对于这样的需求，基于传统搜索引擎很难完成这样一个任务，用户需要逐一打开每个链接并查看其具体信息，毕竟 snippet 并不能提供足够丰富的信息来帮助用户确定该条记录是否是用户所要查找的。Falcons 实体搜索引擎虽然提供了基于实体类型的层次结构来进一步过滤搜索结果，但毕竟功能还比较有限。

图 2 Falcons Explorer 实体搜索界面

Fig.2 Falcons Explore entity search interface

Falcons Explorer 通过基于 Falcons 实体搜索引擎提供的搜索功能，对搜索结果信息进行重新组织，也就是下文将介绍的基于表格和列表的方式来帮助用户快速查找到用户所感兴趣的实体。通过这种方式，可以充分利用已有的基础设施即实体搜索服务和数据检索 API 来帮助用户方便的探索语义 Web 数据。

为了方便用户快速表达查询需求，Falcons Explorer 继承了传统搜索引擎所采用的关键字的搜索方式作为系统开始使用的入口，这样可以大大降低学习系统使用的门槛。如图 2 所示，Falcons Explorer 的界面跟传统搜索界面一样，用户通过输入所要查找实体名称的关键字，搜索之后将自动进入搜索结果页面如图 3 所示，这个过程背后自动调用了 Falcons 实体搜索提供的服务，通过把用户输入关键字发送到 Falcons 实体搜索引擎，然后对返回搜索

结果信息进行重新组织和呈现。

搜索结果页面不同于传统的搜索结果页面，如图 3 所示，主要包含三个视图：1）搜索结果列表视图，默认呈现的视图，即 Overview 视图；2）表格视图，即 Details 视图，提供实体更多维度的信息，并支持基于表格的编程操作，见第 3 节；3）数据源视图，即 Sources 视图，当前实体集合所用到的所有数据源信息。用户可以同时打开多个不同实体集合（可以通过搜索或者导航得到），如图 3 所示，打开了两个实体集合，用户可以在不同实体集合之间切换，对于不再感兴趣的实体集合可以关闭相应的标签页。

图 3 Falcons Explorer 搜索结果页面

Fig.3 Falcons Explorer search result interface

对于列表视图如图 3 的 Overview 视图所示，主要分成左右两部分：1）左半部分为实体集合的类型层次结构，用户可以通过去掉不感兴趣的实体类型或者直接去掉不感兴趣的实例来过滤掉不相关的部分；2）右半部分为返回的实体结果列表，每个实体显示了一些基本信息包括它的标签，URI，评论和图像，如果不存在相应的信息则不会显示，比如评论。

一开始，用户通过输入关键字得到相应的实体集合，由于关键字缺少足够的语义信息，搜索结果可能包含用户不感兴趣的实体，用户可以通过左半部分的类型层次结构的过滤功能，得到感兴趣的实体集合。为了避免一些不相关的实体信息的干扰，用户可以进一步把过滤后的实体作为一个新的集合进行浏览，通过点击列表视图左上角的按钮（“Explore selected categories”）来完成，这时会在当前搜索结果视图下打开一个新的标签页，在该标签页里包含的实体集合为用户过滤后的实体集合。

当用户对于某个实体比较感兴趣想了解更多信息，可以点击该实体，打开一个单独的实

体视图，如图 4 所示。由于 Falcons Explorer 背后的数据是来自于真实的 Web，因此一个实体的描述信息往往来自于不同的数据源，该实体视图会把来自于不同的数据源的数据进行融合，通过 Falcons 提供的 API 来异步请求实体的信息。通过这种异步加载的方式，一方面可以避免由于数据量过多等待时间过长，导致页面出现空白的效果；另一方面根据每个数据源 URI 来异步获取其相应的数据，将获取到的数据实时更新到视图，可以带来比较好的用户体验。实体信息按照实体的属性进行组织，用户可以通过右边的属性列表快速导航到感兴趣的实体属性信息。该实体视图提供了三种导航方式：1）用户可以点击实体视图左上角的“Explore in the collection perspective”链接把当前浏览的实体作为一个单一实体的集合来浏览，进入如图 3 所示的实体集合视图；2）用户可以点击任何属性值为实体的超链接，打开一个类似于图 4 的新的实体视图；3）用户可以把实体属性的所有值作为一个集合来浏览，通过点击属性最右边的放大镜按钮。此外，用户可以折叠或者展开相应的属性甚至是所有属性的面板，通过点击“Sources”链接可以查看该实体所有用到的数据源信息。

图 4 浏览单一实体信息

Fig.4 Browsing a single entity

3 基于表格视图编程

由于传统的基于列表的方式所提供的信息和功能非常有限，不利于用户进一步探索搜索结果的信息，因此也很难在搜索结果基础上做进一步的查询表达。为了查找到用户感兴趣的实体，用户需要逐一打开每个实体。基于这种传统的方式，在信息探索上比较低效，不利于用户快速查找到感兴趣的信息。

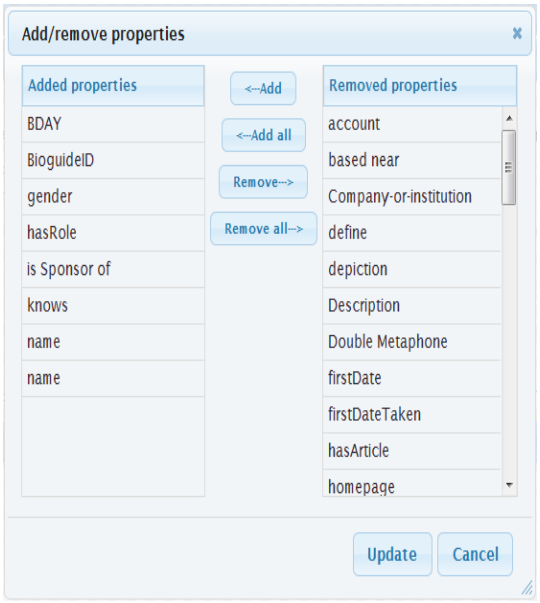
为了解决这个问题，Falcons Explorer 对于搜索结果，提供了一个基于表格的视图如图 5（a）所示，每一行代表一个实体，每一列代表实体的一个属性，通过每个单元格，可以浏览到每个实体相应属性值的信息，由于实体的数据源并不唯一，因此每个单元格包含的值可能多个（对于包含多个值的单元格，会有相应“Show all”链接），可以通过点击单元格的“Show all”查看更多值。由于实体的属性可能比较多，默认情况下不会全部显示，只会随机显示其中 10 个属性。用户可以通过该表格视图，快速找到感兴趣的实体，同时可以比较

直观的对实体进行比较，该表格视图支持面向模式（schema）和数据的编程，可以帮助用户完成复杂查询的表达，3.1 节和 3.2 节将分别介绍面向模式和数据的编程，3.3 节对面向高级用户提供的基于关系视图的编程做了简介，3.4 节介绍了如何在表格视图进行导航。

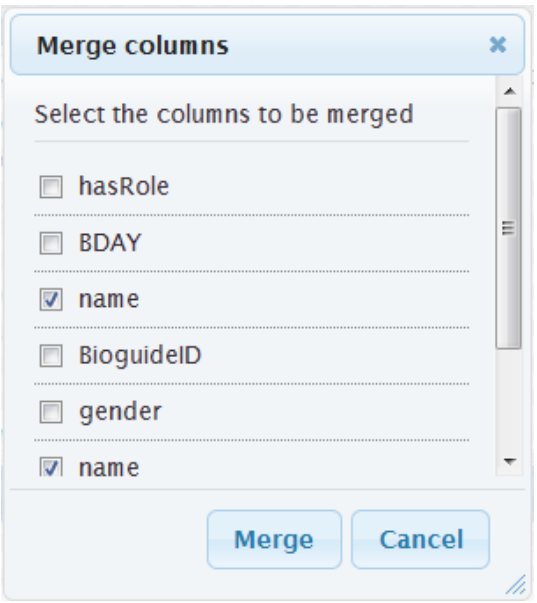
3.1 面向模式编程

实体的属性通常不止一个，而且并不是每个属性都是用户感兴趣的，比如浏览一组人的信息的时候，可能用户当前只对其工作信息比较感兴趣，对其它信息可能不感兴趣，面向模式的编程提供了添加/删除列（属性）的功能，帮助用户只选择感兴趣的实体属性进行浏览，通过如图 5（a）工具栏的“Add/remove properties”按钮可以得到图 5（b）所示的对话框，用户可以选择想要呈现和不想要呈现的列。

除了通过对列进行添加或者删除操作之外，面向模式的编程还支持对不同列进行合并以产生一个新的列，这对于实体的信息来自于不同的异构的数据源是非常有用的一个操作，比如对于人的信息，往往有不同的本体定义了类似或相同的属性概念来描述人的信息，比如描述人名字信息的属性有 foaf:name，rdfabout:name 等，通过把这些等价的概念属性合并成一个新的列，可以比较方便地完成异构数据源的融合，同时也有助于去除冗余的信息。通过如图 5（a）工具栏的“Merge columns”按钮可以得到图 5（c）所示的对话框，通过选择所要合并的列比如这里的两个 name 属性，选择合并之后，这两列将合并成一个新的列“name/name”（新列的标签由原始列的标签所组成，以左斜线分割）。当完成列的合并操作后，用户也可以选择将其重新分离，通过该列表头的下拉菜单选项“Split this column”来完成。对于列的合并，自定义列（由其它列合并得到）也可以用于合并产生新的列。以上这些操作的组合，可以很好帮助用户聚焦到自己感兴趣的属性，同时也有利于实现数据融合和冗余信息的去除，帮助用户更好去探索实体信息。

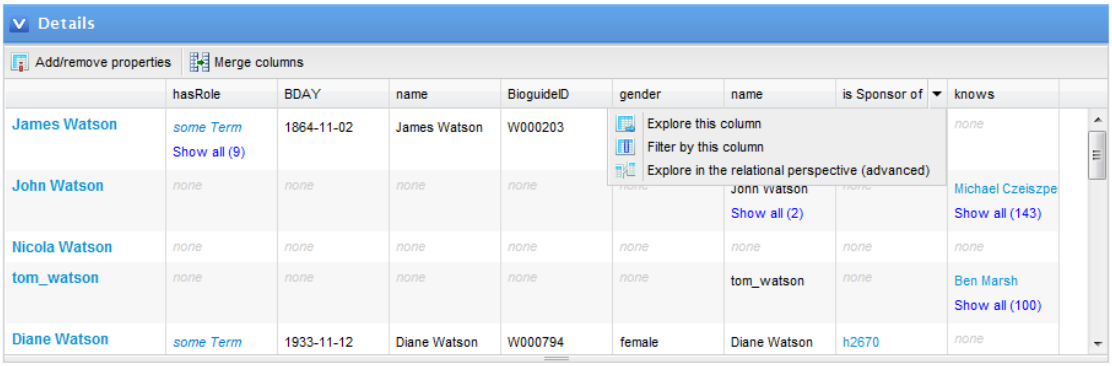


(b)



(c)

(d)



(a)

图 5 实体集合详细信息。(a) 将所有 RDF 三元组组织成表格，每行代表一个实体，每列代表一个属性，单元格包含属性值；(b) 为表格添加/删除列；(c) 合并列；(d) 根据表格列的值对行进行过滤。

Fig.5 Details view of entity collection. (a) Related RDF triples are organized into a table, where each row stands for an entity, each column stands for a property, and the cells hold property values. (b) Add or remove properties from the table. (c) Merge columns. (d) Use property values as well as their categories for filtering.

3.2 面向数据编程

为了根据实体的属性快速找到用户感兴趣的实体，面向数据的编程允许用户根据列的属性值对表格行进行过滤，用户可以选择想作为过滤条件的列，通过点击如图 5（a）下拉菜单的“Filter by this column”这个选项，可以得到图 5（d）的过滤对话框，Falcons Explorer 会根据属性值的类型形成一颗基于属性值的类型所构成的层次结构，用户可以根据属性值的类型或者具体的属性值作为过滤条件。当创建了一个过滤条件后，将动态计算满足该过滤条件的行（实体），对于不满足的行会移除掉，这样用户可以快速找到感兴趣的实体。此外，同时为多个列（属性）创建过滤条件是允许的，列与列之间的限制条件满足与的关系。

结合前面提及面向模式编程，可以帮助用户构建出相对复杂的查询。此外，基于表格的查询构建过程是和实体信息探索过程相结合。用户通过实体信息探索可以构建出更有效的查询，这种面向实例数据查询的构建相比于直接输入 SPARQL 这种结构化查询语言，更能让普通用户所接受，同时又具有一定的查询表达能力。

3.3 基于关系视图编程

除了基于表格视图所提供的编程操作外，Falcons Explorer 还为高级用户提供了更强的查询表达能力，基于关系视图的编程方式。如图 5（a）所示通过选择感兴趣的列，选择其下拉菜单的“Explore in the relational perspective(advanced)”菜单项，可以进入相应的关系视图编程界面，基于关系视图编程要求用户具有一定的数据库方面知识，Falcons Explorer 为关系视图编程提供了投影（Projection）、联接（Join）和 Tie[5]操作。

3.4 改变浏览焦点

用户浏览过程中，往往会在不同实体集合间切换，为了更好支持用户浏览过程中焦点的变化，除了前面列表视图提及的方法，表格视图也提供了一些相应的功能来支持这种焦点变化。用户在基于表格视图探索过程中，除了可以访问感兴趣的实体（通过打开一个新的实体视图页面）外，用户还可以通过列的操作“Explore this column”（如图 5（a）下拉菜单所示），来改变浏览的焦点，Falcons Explorer 会把该列相应的所有属性值作为一个新的实体集合在当前实体集合视图下重新打开一个新的实体集合视图标签页。通过切换标签页，可以返回原来浏览的实体集合，对于不再感兴趣的实体集合，用户可以将其相应的标签页关闭。

除了以上提及的切换浏览焦点的方式外，还可以直接通过实体集合视图上方的搜索框，通过输入新的实体名称来得到一个新的实体集合，相应的也是打开一个新的实体集合视图标签页而不会覆盖当前的实体集合视图。

4 结论

本文给出了一种结合实体搜索和表格式编程的方法，该方法以基于关键字的实体搜索作为系统的入口，对传统搜索结果视图进行改进，增加了表格视图来增强用户探索语义数据和表达查询的能力。此外，基于表格视图的编程也提供了一种实现异构数据源融合和数据冗余消除的方案。基于本文提出的方法，我们构建了 Falcons Explorer 系统，该系统为用户提供了有效的探索和查询语义 Web 数据的能力。

在未来工作中，主要会针对当前系统做进一步的评估，针对当前系统所存在的不足进行改进以完善该系统。

致谢

感谢程龚老师和龚赛赛同学在系统开发中所给予的帮助和支持，也感谢瞿裕忠老师的建议和指导。

[参考文献] (References)

[1] C. Bizer, T. Heath and T. Berners-Lee. Linked data - The story so far. International Journal on Semantic Web and Information Systems, 2009, 5(3):1-22.

[2] Cheng, G., Qu, Y. Searching Linked Objects with Falcons: Approach, Implementation and Evaluation. International Journal on Semantic Web and Information Systems, 2009, 5(3): 49-70.

[3] Zenz, G., Zhou, X., Minack, E., Siberski, W., Nejdl, W. From Keywords to Semantic Queries - Incremental Query Construction on the Semantic Web. J. Web Semant., 2009, 7(3): 166-176.

[4] Scaffidi, C., Shaw, M., Myers, B. Estimating the Numbers of End Users and End User Programmers. In: 2005 IEEE Symposium on Visual Languages and Human-Centric Computing, pp. 207-214. IEEE Computer Society, Washington, DC, 2005.

[5] Codd, E.F. A Relational Model of Data for Large Shared Data Banks. Commun. ACM, 1970, 13(6): 377-387.