计算机科学2005Vol.32№.3

基于语义的Web挖掘

伏 晓 骆 斌 陈世福
(南京大学计算机软件新技术国家重点实验室 南京210093)
(南京大学计算机科学与技术系 南京210093)

摘 要 基于语义的Web挖掘是使用从现有Web数据中抽取的语义或直接使用Web数据中已有的语义结构来帮助Web挖掘。它有效地结合了语义网和Web挖掘两个领域的研究成果,既可以通过开发新的语义结构来帮助Web挖掘,又可以利用挖掘结果促进语义网的创建。本文介绍了基于语义的Web挖掘的基本思想和研究现状,分析了语义网和Web挖掘相结合的优势,并详细论述了国际上关于利用数据挖掘技术创建语义网,利用语义挖掘Web数据和直接挖掘语义网三个方面的研究工作。

关键词 语义网,Web挖掘,XML,RDF,本体

Research of Semantic Web Mining

FU Xiao LUO Bin CHEN Shi-Fu
(State Key Laboratory for Novel Software Technology,Nanjing University,Nanjing 210093)
(Department of Computer Science and Technology,Nanjing University,Nanjing 210093)

Abstract This paper gives an overview of the main ideas and the research progress of Semantic Web Mining. Semantic Web Mining utilizes the semantic data extracted from the traditional Web or uses the semantic structure of the Semantic Web directly to help Web Mining. It combines two fast-developing research areas Semantic Web and Web Mining. This idea is to improve,on the one hand,the results of Web Mining by exploiting the new semantic structures in the Web; and to make use of Web Mining,on the other hand,for building up the Semantic Web. In this paper,how to extract semantics from the Web is introduced firstly. Then exploiting Semantics for Web Mining and mining the Semantic Web directly is discussed.

Keywords Semantic Web,Web Mining,XML,RDF,Ontology

1 引言

当前WWW的发展速度极为惊人,整个网络正在形成一个前所未有的超级信息数据库,这使得如何处理这些海量信息成为了一个全新的课题。然而,目前的网络搜索引擎平均只能检索25%的可获取信息,其返回结果经常会包含大量的无用信息,这就为人们寻找所需信息带来了很大的困难。另一方面,由于Web数据大部分是非结构化的,这就导致传统数据挖掘技术对Web进行挖掘的效果总是不尽如人意。造成这些问题的一个重要原因是大量的Web数据只能人工解析,机器自动处理的能力很弱。因此Tim Berners-Lee提出了语义网(Semantic Web)的思想[1],即Web上定义的链接数据不仅能够显示,而且还应该是机器可理解的,也就是说可以被机器自动地处理、集成和重用。

由于Tim Berners-Lee提出的这种新的语义结构能够有效地改善Web挖掘的结果,于是一些学者将其引入Web挖掘领域,从而形成了一个新的研究领域:基于语义的Web挖掘。基于语义的Web挖掘的目标是利用从现有Web数据中抽取的语义或直接利用Web数据中已有的语义结构来帮助Web挖掘。它将两个发展迅速的领域:语义网和Web挖掘结合了起来,并且利用了这两个领域的研究成果。因此,基于语义的Web挖掘一方面可以通过开发新的语义结构来帮助Web挖掘,另一方面还可以利用挖掘结果促进语义网的创建。

本文主要介绍了基于语义的Web挖掘的一些基本思想和研究现状,讨论了语义网和Web挖掘的结合究竟会带来哪些益处。本文将从利用数据挖掘技术创建语义网,利用语义挖掘Web数据和直接挖掘语义网三方面展开。

2 Web挖掘概述

Web挖掘是一个极其复杂的过程,它不同于传统的数据仓库技术和简单的知识发现(KDD),它面对的海量信息不全是简单的结构化数据,而常常为半结构化的数据,如文本、图形、图像数据,甚至是异构型数据。发现知识的方法可以是数学的,也可以是非数学的;可以是演绎的,也可以是归纳的。

Web挖掘通常被分为三类:Web内容挖掘(Web content mining),Web结构挖掘(Web structure mining)和Web使用记录(Web usage mining)的挖掘[2]。另外,Web结构也可以被认为是Web内容挖掘的一部分。

Web内容挖掘[2]是从Web的内容、数据或文档中发现有用信息的过程。从信息检索的角度来看,Web内容挖掘可以促进信息查询,帮助过滤信息;从数据库的角度来看,Web内

伏 晓 硕士研究生,研究方向为数据挖掘,语义网络。骆 斌 教授,博士,研究方向为数据库,人工智能。陈世福 博士生导师,研究方向为人工智能。

容挖掘可以通过信息集中、建模来帮助查询。从资源形式看，网络信息内容是由文本、图像、音频、视频、元数据等形式的数据组成的，因此 Web 内容挖掘是一种多媒体数据挖掘形式[3]。

Web 结构挖掘[4]即挖掘 Web 潜在的链接结构模式。这种思想源于引文分析，即通过分析一个网页链接和被链接数量以及对象来建立网络自身的链接结构模式。可以用于网页归类，并且可以由此获得有关不同网页间相似度及关联度的信息，有助于用户找到相关主题的权威站点。

Web 使用挖掘[5]可以帮助了解用户的网络行为数据所具有的意义。Web 内容挖掘、Web 结构挖掘的对象是网上的原始数据，而 Web 使用挖掘则面对的是在用户和 Web 交互的过程中抽取出来的第二手数据。这些数据包括：Web 服务器访问记录、代理服务器日志记录、浏览器日志记录、用户简介、注册信息、用户对话或交易信息、用户提问式等等。

3 创建语义网

Tim Berners-Lee 提出语义网必须在不同的层次构建[1]：Unicode/Unified Resourse Identifier，XML，RDF，Ontologies，Logic，Proof，Trust 如图1所示。

Unicode 和 URI(Uniform Resource Identifier)是语义网的基础，其中 Unicode 处理资源的编码，URI 负责标识资源。XML(eXtensible Markup Language)则是为 Web 数据提供一种好的语法描述。使用 XML 可以在计算机之间方便地解析各种类型的数据。XML Schema 是用来定义数据结构。RDF(Resource Description Framework)提供了一个标准模型来描述 Web 资源，RDF Schema 增强了 RDF 对资源的描述能力，其作用类似 XML Schema 之于 XML，用于定义资源的类型。本体(Ontology)是一套知识术语集，包括词汇(Vocabulary)、语义关联和一些简单的逻辑推导规则。它与概念在各个专门领域的实际应用(即实用数据)一起构成语义网的基础，可以描述各种资源之间的联系。Logic 则是构建推导引擎的理论基础。通过推导引擎可以从已有知识中推出新的知识。Prove 和 Trust 则是面向语义网的信用评估和安全问题，随着语义网的发展，这些问题必将更加突出。这七层中核心层是 XML、RDF、和 ONTOLOGY，Web 信息的语义主要由这三层表示。

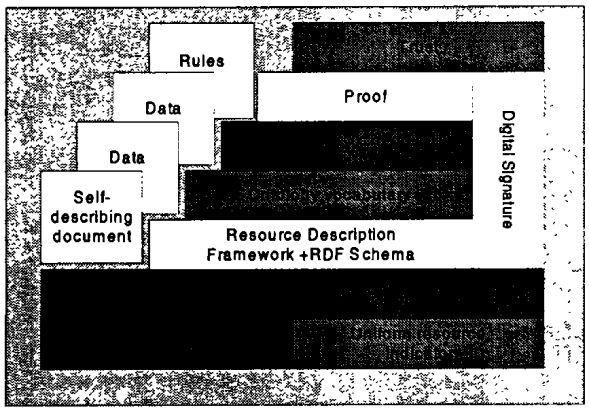

图1 Semantic Web 的层次结构

3.1 基于 XML 和 RDF 创建语义网

XML 是计算机可读文档的规范，它让每个人都能创建自己的标签来对网页或页面的部分文字进行注释[6]。它允许用户在文档中加入任意的结构，但无需说明这些结构的含意。通过 DTD(Document Type Definitions)和 XML Schema 可以定义数据结构。

基于 XML 创建语义网需要以下技术：根据 XML 的语义来计算 XML 文档的相似度，并以此来预处理用于 XML 数据挖掘的 XML 文档；使用序列模式挖掘算法挖掘出满足用户定义的最小支持度的 XML 结构之间的最相似路径。

现有的 XML 的语义匹配系统有：TransScm system[7]、LSD system[8]和 Xmapper[9]等。TransScm system 通过使用标定图(labeled graphs)来匹配模式(schema)，这些模式建立在从 DTD 文件中抽取出的结构和标记名基础之上。LDS system 采用多策略学习，它利用机器学习的算法通过用户自定义的映射表来发掘用于匹配的映射表。Xmapper system 仅使用独立的 XML 文档来产生映射：使用机器学习来提高这些映射表在不同领域的准确性。

RDF 是在使用 XML 定义网络方面迈出的第一步。XML 提供了一个为数据编码的方式，而 RDF 则能够说明数据本身，也就是语义。它不是一种语言，而是一个表达网上数据的模型。所以也被称作“元数据”。RDF 的基本数据模型包含三类对象：资源(resource)、属性(property)和陈述(statement)[10]。资源就是指网络上的数据。属性用来描述资源的一个方面、特征、属性以及关系，陈述则用来表示一个特定资源，它包括一个命了名的属性和它对应资源的值，所以一个 RDF 描述实际上就是一个三元组：

(object[resource], attribute[property], value[resource or literal])

例如，表1给出了一个叫 John 的人(有个电话号码)创建了一个指定的网页的 RDF 描述。

表1 对于 John 创建了一个指定的网页的 RDF 描述

OBJECT	ATTRIBUTE	VALUE
http://www.w3.org/	created_by	#anonymous_resource
#anonymous_resource]	name	“John”
#anonymous_resource]	phone	“47782”

可以很容易地将一个 RDF 描述转为一个标记图。其中用椭圆表示资源，用箭头表示属性，而用方框表示 litral 值。图2是对上面例子的图表示。

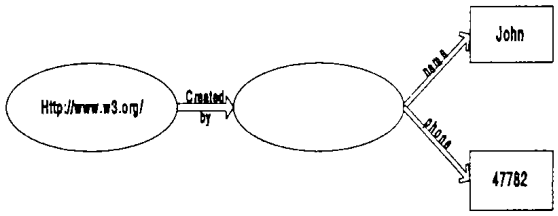

图2 对于 John 创建了一个指定的网页的 RDF 描述

由上例可见 RDF 只是一个数据模型，它本身对于语法是无知的，因此可以标记图、二维表等方式来表示它，而 XML 显然也是一种很好的表示方式。又由于 RDF 只是提供了一个用于领域无关的机制来描述元数据，因此它需要 RDF Schema 来辅助定义领域相关的 properties 以及用于使用这些 properties 的 resources 类。RDF Schema 实际上就是 RDF 的类型系统。

3.2 基于本体(Ontology)创建语义网

本体现在已经成为很多领域中的一个流行课题。它最早是一个哲学上的概念,是有关存在的本质,以及何种事物存在的理论。后来,人工智能和网络研究人员也选择了这个词作为其术语。在他们看来,本体是一份描述正式定义名词及它们之间关系的文档或文件。

Gruber 曾给出本体的一个被广泛接受的定义[11],即"本体是概念模型的明确的规范说明"。后来,Borst 在此基础上给出了本体的另外一种定义[12]:"本体是共享概念模型的形式化规范说明"。这包含4层含义[13]:概念模型(conceptualization)、明确(explicit)、形式化(formal)和共享(share)。"概念模型"指通过抽象出客观世界中一些现象(Phenomenon)的相关概念而得到的模型。概念模型所表现的含义独立于具体的环境状态。"明确"指所使用的概念及使用这些概念的约束都有明确的定义。"形式化"指本体是计算机可读的(即能被计算机处理)。"共享"指本体中体现的是共同认可的知识,反映的是相关领域中公认的概念集,即本体针对的是团体而非个体的共识。本体的目标是捕获相关领域的知识,提供对该领域知识的共同理解,确定该领域内共同认可的词汇,并从不同层次的形式化模式上给出这些词汇(术语)和词汇间相互关系的明确定义。根据领域依赖程度的不同,本体可以分为上层本体(top-level ontology)、领域本体(domain ontology)、任务本体(task ontology)和应用本体(application ontology)。用于表示本体的语言有很多种,如 SHOE、Topic Maps、XOL、RDF 和 RDFS 以及 DAML+OIL 等。这些语言一般都具有域内类的继承性描述、类的属性和类的实例等概念。

基于本体创建语义网需要以下技术:从网络中抽取本体,即本体学习(ontology learning);本体的集成,即本体的映射(mapping)及合并(merging)。

本体学习涉及到数据挖掘以及自然语言处理等技术,其过程是以下主要步骤的循环:导入/再利用(import/reuse)、抽取(extract)、修剪(prune)、应用(apply)。本体学习的结构如图3所示。自动实现本体学习的系统有 Text-to-Onto,它是一个从文本自动学习本体的系统。

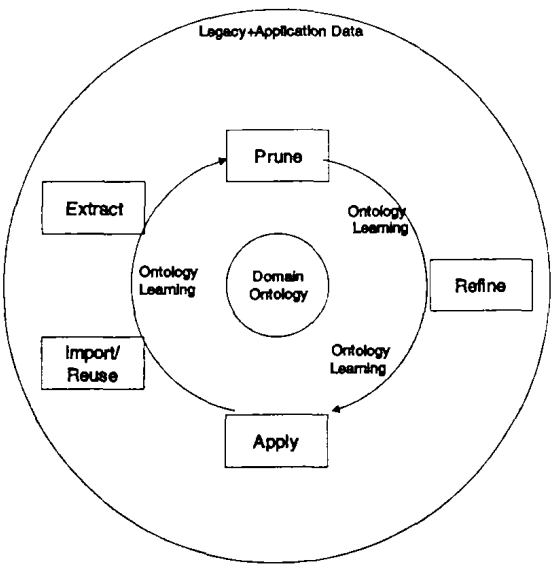

图3 语义网本体学习结构

实现本体合并的算法有很多,比较著名的 FCA-MERGE 是一种自底向上的本体合并算法。本体映射需要评估本体间的相似度、对本体进行标准化分析以及发现和说明不同版本的本体间的关系。

3.3 XML Topic Maps

在传统的 XML 和 RDF 之外 BenedicteLe Grand 等人又提出了一种新的语义网络结构 XTM(即 XML Topic Maps)[14]。Topic Maps 是类似书籍目录的一种结构,是一种管理连接信息的有效方法。这种方式允许用户创建大量的元数据和与之紧密相关的数据。Topic Maps 能和 RDF 互相转化。一个 Topic Maps 是一个 topic 和 topics 之间关系的集合。Topic Maps 通过外部指针连接这些 topic,就像 URL 用于连接资源。用这种方式组织数据更加有利于网络数据的检索。这主要通过以下几个方面实现:

·通过定义 topic maps 和站点的 profiles(它主要用于描绘 Web 站点的特征),可以在语义标准的基础上来衡量 Web 站点和用户需求之间的关系。

·通过过滤 topic maps 可以滤去 sigular topics,也就是说可以从语义上鉴定并清除"无趣"的主题。

·通过集合概念相关的 topics 以及通过对不同等级或层次的细节的可视化来促进对 Web 的浏览。

现在已经有一些软件可以支持创建和浏览 Topic Maps,如 Ontopia navigator。

4 利用语义帮助 Web 挖掘

利用语义帮助挖掘可以在很多方面改善传统的 Web 数据挖掘。在内容挖掘方面,语义能够带来对文档的内容和含义更清楚的认识;在结构挖掘方面,语义意味着更清晰的结构;在使用挖掘方面,利用语义可以获得与用户行为更为相关的信息。

4.1 利用语义挖掘 Web 内容

在 Web 内容挖掘过程中可以使用诸如本体之类的背景知识来帮助预处理,从而改善聚类结果并且帮助选择聚类结果。例如 A. Hotho 曾提出一种基于本体的文本聚类方法[15]:COSA(Concept Selection and Aggregation),其基本思想是用一个简单的核心本体来限制相关文档集的特征,以及来自动地产生好的集成。聚类算法仍然是标准的 K-Means,但是利用了基于本体的启发式搜索算法。其基本步骤是:首先使用一个有效的自然语言处理系统将单词映射到概念;然后使用概念层次结构为下一步的聚类生成好的聚集。COSA 的基本策略包括计算单一概念的支持度以及自顶向下贪心式层次遍历算法。采用 COSA 不需要人工干涉便能自动产生结果,然后用户可以像使用传统的性能衡量标准一样,根据用于聚类的概念来决定采用哪种聚类结果。同时,还可以用本体中的相应概念来描绘和解释聚类结果。

在利用语义帮助 Web 内容挖掘方面的应用还有不少,例如可以将语义引入到邮件过滤系统中。人们常常会受到大量广告之类的垃圾邮件的困扰,尽管多数邮件服务系统都提供了过滤功能,但是一些恶意邮件通过伪装标题仍然能够轻易地绕过过滤系统。如果在过滤邮件时利用语义,也就是根据邮件的内容来设置挖掘条件,那么过滤的准确性显然能够大大提高。

4.2 利用语义挖掘 Web 结构

如果在挖掘 Web 结构的时候把页面的内容考虑在内,那么结构挖掘的结果就会得到很好的改善。著名的 PageRank 算法把任何被"大量引用"的页面都视为"相关页面",而没有

考虑该页面的内容是否与查询有关。因此如果把超链的文字内容和超链的环境相结合进行挖掘，得到的结果将更为精确，这正是 CLEVER[16]所做的。Focused Crawler[17]在这方面也作了一些尝试，他把主题的内容和链接图相结合，并且采用了更为灵活的挖掘方法。随后又有一些学者把语义的思想引入进来，在 Focused Crawler 的基础上建立了一种基于本体的 Focused Crawling 方法[18]。其基本思想是：一个 Web crawler 驻留在一台机器上，通过 Internet 简单地向其他机器发送针对文档的 HTTP 请求。它的一个基本的应用是作为搜索引擎的索引。Marc Ehrig 曾提出一个基于本体的 crawling 框架[19]（如图4），在该框架中 crawling 的过程包括两个互相联系的循环：本体循环（the Ontology cycle）和 crawling 循环（the crawling cycle）。本体循环主要由人工驱动，其工作包括以本体实例的形式定义 crawling 的目标，并且以文档列表的形式将 crawling 过程的输出提供给用户（其目的是改善用户定义的现有本体）。Crawling 循环构成了 Internet 上的 Crawler。它与网络上的数据自动交互，并且对其进行检索，然后与本体联系以确定这些数据的相关度。相关度计算主要是用来为用户选择相关文档，以及关注那些能够用于未来搜索相关文档以及 Web 元数据的链接数据。

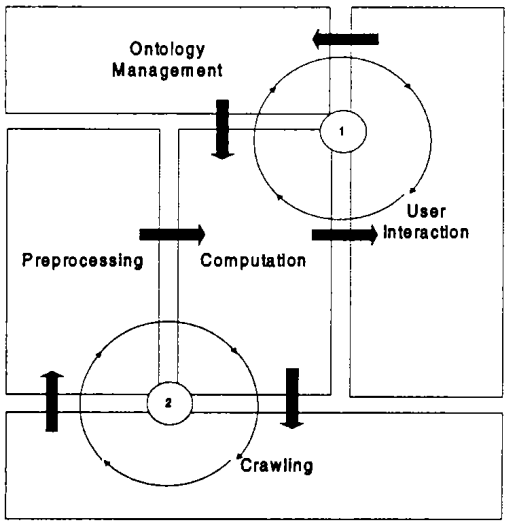

图4 Ontology-Focused Crawling 的过程及其与系统结构元素的关系

4.3 利用语义挖掘 Web 使用

利用用户访问路径上的页面的语义可以有效地改善网络使用挖掘的结果，因为语义可以帮助分析者理解什么是用户所寻找的，以及哪些内容经常会同时出现。在给一个 Web 站点上的大量页面分类时，最基本的方式是可以根据站点的全局本体模型，利用人工生成的本体结合自动模式来进行分类。例如，有一个基于关系数据库的 Web 站点，站点上包括一定数量的静态页面和大量根据用户的查询请求动态生成的页面。可以用一个本体来描述整个站点，而且通过把用于产生动态页面的查询字符串映射到相应的概念可以生成一个分类模式。

根据内容（即页面描述的对象类型）、结构（即页面在对象查询中所起到的作用）以及服务（用户所选择的查询功能的类型）将概念构造成一个层次结构。页面通过与该层次结构建立映射来分类。这样一个用户的浏览路径就可以看作概念层次结构中一系列概念的集合。这种分类方式能够使 Web 使用挖掘的结果更加合理，Web 站点的可重构性更强、Web 站点的个性化也更好——因为语义分析有助于辨别用户的行为模式，区别出成功完成一个寻找过程的用户和中途退出的用户[20]，这样就有利于动态地为新用户产生帮助信息。

对于没有显式的包含语义信息的页面，可以首先利用一般的信息检索技术通过关键字分析自动抽取页面内容。使用路径可以根据公共内容聚类。这有助于分析者理解哪些信息是用户频繁搜索的，还可以辨别出在用户日志中经常同时出现的内容，并据此生成参考信息。

总而言之，借助语义的 Web 使用挖掘，不仅能更好地理解用户的浏览路径，还能给在线用户提供帮助。S. Parent 就曾提出过一种将信息检索技术、本体、挖掘用户日志几项技术结合起来以提高查询效率的方法[21]。

5 挖掘语义网

对于传统结构的 Web 页面，可以通过前文所述的方法首先抽取语义，然后利用语义进行网络挖掘。当然，也可以直接挖掘现成的语义网。与挖掘传统网络一样，可以把语义网挖掘分为三类：语义网内容挖掘、语义网结构挖掘和语义网使用记录挖掘。

5.1 语义网内容和结构挖掘

在语义网中，内容和结构是紧密相连的，因此内容挖掘和结构挖掘之间的界限也很模糊。

语义网内容/结构挖掘中采用的一个重要的技术叫做关系数据挖掘（Relational Data Mining），又名归纳逻辑编程（Inductive Logic Programming）。关系数据挖掘在关系数据库中寻找涉及多个关系的模式，它包括了分类、回归、聚类和关联分析等多项技术。它能够直接地转化算法，因此能够处理用 RDF 或本体表示的数据。在这方面有两个问题需要考虑，第一是被处理的数据的尺寸，也就是算法的可伸缩性。第二是数据分布在整个语义网络，没有一个中心数据库服务器这样一个事实。在可伸缩性方面，ILP 算法还需要进一步改进。而对于第二个问题，一种好的思想是只传输中间结果而非整个数据集。

5.2 语义网使用记录挖掘

如果通过参考本体中的概念使语义被显示地包含在页面中，使用挖掘就可以得到很大的提高。语义网使用记录挖掘可以在基于本体创建的日志文件上进行[22]。挖掘这样的日志文件能够有效地建立相同兴趣用户的聚集，从而为用户提供基于本体的个性化视图。

结束语 本文涉及两个迅速发展的领域：语义网和 Web 挖掘，主要研究了这两个领域的结合——基于语义的 Web 挖掘。讨论了如何利用 Web 中的语义结构改善 Web 挖掘的结果，以及如何利用 Web 挖掘技术构造语义网。

随着网络的飞速发展，传统 Web 在信息显示和处理上的不足之处也更加显著。而语义网作为一种新型的网络结构，较好地克服了这些问题。正如 Tim Berners-Lee 所描绘，语义网代表着网络的未来，必将成为下一代互联网的神经。而基于语义的 Web 挖掘作为与这一趋势相适应的技术，也必将成为 Web 挖掘研究的新热点。当前，在如何基于传统 Web 创建语义网方面的技术已经日趋成熟，在如何利用语义帮助 Web 挖掘方面也已经有了比较多的研究。而在如何直接挖掘语义网方面相关工作还不是很充分，相信随着语义网的发展，这一领域也会更加成熟。

参考文献

1 Berners-Lee T. A roadmap to the Semantic Web. http://www.w3.org/DesignIssues/Semantic.html Sept. 1998
2 Kosala R, Blockeel H. Web Mining Research: A Survey. ACM-SIGKDD, July 2000
3 Zaiane O R, et al. Multimediaminer: a system prototype for multimedia data mining. In: Proc. ACM SIGMOD Intl. conf. on. Management of Data, 1998. 581～583
4 Chakrabarti S, et al. Mining the link structure of the world wide web. IEEE Computer, 1999, 32(8): 60～67
5 Cooley R, Mobasher B, Srivastava J. Web mining: Information and pattern discovery on the world wide web. In: proc. of the 9th IEEE Intl. Conf. on Tools with Artificial Intelligence(ICTAI'97), 1997
6 Bray T, et al. Extensible Markup Language (XML) 1.0 (Second Edition), W3C Recommendation. Oct. 2000. http://www.w3.org/TR/2000/ REC -xml-20001006
7 Milo T, Zohar S. Using schema matching to simplify heterogeneous data translation. In: proc. 24th Int Conf on Very Large Data Bases, 1998. 122～133
8 Andersen E S, Valente D M. the art of simulation and the Lsd system, ETIC Session, Strasbourg, 2003
9 http://www.nalasoftware.com/DOCS/Xmapper/overview.cfm
10 Decker S, et al. The Semantic Web: The Roles of XML and RDF. IEEE INTERNET COMPUTING 1089-7801/00/$10.00 (c) 2000 IEEE, 2000, 4 (5): 63～74
11 Gruber T R. A Translation Approach to Portable Ontology Specifications. Knowledge Acquistion, 1993, 5: 199～220
12 Borst W N. Construtions of Engineering Ontologies for Knowledge Sharing and Reuse: [PhD thesie]. Univerity of Twente, Enschede, 1997
13 Studer R, Benjiamins V R, Fensel D. Knowledge Engineering, Principles and Methods. Data and Knowledge Engineering, 1998, 25(1-2): 161～197
14 Grand B L, Soto M, Dodds D. XML Topic Maps and Semantic Web Mining
15 Hotho A, Maedche A, Staab S. Ontology-based text clustering. In: Proc. of the IJCAI-2001 Workshop "Text Learning: Beyond Supervision", August, Seattle, USA, 2001
16 Chakrabarti S, et al. Automatic resource compilation by analyzing hyperlink structure and assciated text. In: Proc. of the 7th World-wide web conf. (WWW7), 1998, 30(1-7): 65～74
17 Chakrabarti S, van den Berg M, Dom B. Focused crawling: Anew approach to topic-speci. c web resource discovery. In: Proc. of the 8th World-wide web conf. (WWW8), 31(11-16), Toronto, May 1999. 1623～1640
18 Maedche A, et al. Ontology-focused crawling of documents and relational metadata. In: Proc. of the Eleventh Intl. World Wide Web Conf. WWW-2002, Hawaii, 2002
19 Ehrig M. Alexander Maedche Ontology-Focused Crawling of WebDocuments
20 Spiliopoulou M, Pohle C. Data mining for measuring and improving the success of web sites. Data Mining and Knowledge Discovery, 2001, 5
21 Parent S, Mobasher B, Lytinen S. An adaptive agent for web exploration based of concept hierarchies. In: Proc. of the 9th Intl. Conf. on Human Computer Interaction, New Orleans, LA, 2001
22 Hotho A, Maedche A, Staab S, Studer R. SEAL-II - the soft spot between richly structured and unstructured knowledge. Journal of Universal Computer Science (J. UCS), 2001, 7(7): 566～590

(上接第168页)

DSPs 间或在 GPPs 间，则以主频的高低决定性能的优劣。GPPs 在 DSP 类程序上的劣势稍低，是因为 GPPs 为了满足嵌入式领域的发展需求，而在指令集或者硬件结构上进行了增强以对 DSP 算法提供支持。主频高的 DSPs 性能比主频低的 GPPs 性能稍好，说明所采用的体系结构和主频的高低对性能的优劣也起着一定的作用。

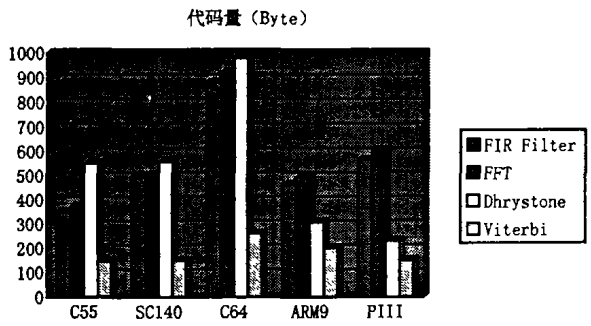

图4 代码量的评测与比较

从图4代码量的测试中可以看出，处理器的结构对于代码量的大小起着决定性的作用。VLIW 结构的采用，增加了指令长度，从而增大了代码量。Superscalar 结构的采用也适度增加了代码量。代码量的大小与编译器的选择有关。从图中可以看出，对于 DSP 类程序，DSPs 的代码量小于 GPPs；对于控制类程序，GPPs 的代码量小于 DSPs。这是因为针对 DSPs 的编译器在代码优化和调度时，其算法倾向于最大限度地使指令并行，并最大限度地利用 DSPs 的硬件资源。控制类程序不具备 DSP 算法的特点，编译器在对代码进行分析时，其优化算法和调度算法无法发挥最大效能，从而不能较好地对代码进行优化和调度，从而代码量较大。对于 GPPs 亦然。

结束语 综上所述，由于 GPPs 和 DSPs 面向的应用不同，所具有的功能不同，从而导致了 GPPs 和 DSPs 从指令集结构、体系结构到存储器结构都具有显著的不同。DSPs 在对运算密集型应用的支持上，从性能到代码量的大小，都具有明显的优势，但是对控制密集型应用的支持则比较有限；而 GPPs 则对控制密集型应用提供良好的支持，对于运算密集型应用的支持不如 DSPs。

随着应用的推动和处理器体系结构研究的发展，GPPs 开始考虑在有效支持控制密集型应用的同时，也能够有效支持运算密集型的应用。高性能通用微处理器开始借用 DSP 处理器的结构优点，并在硬件上通过协处理器或者增加 DSP 功能单元的方式对 DSP 功能提供支持。但由于 GPPs 缺乏实时可预测性，存储器带宽有限、优化 DSP 代码困难，有限的 DSP 工具支持，高功耗等问题，因此单纯将 GPPs 应用于 DSP 领域，功能和性能都不能满足要求。而 DSP 处理器也不能单纯地局限于数字信号的处理，而也应具有高效的控制能力。但其独特的功能和结构、对非 DSP 任务进行支持的第三方工具的缺乏决定了它对控制密集型应用的支持非常有限。将通用微处理器的体系结构和 DSP 处理器体系结构进行有效的融合，生成融合型的高性能微处理器，成为嵌入式处理器发展的主流趋势。在嵌入式应用领域中，将高性能嵌入式通用微处理器和 DSP 处理器进行融合，形成专用的融合型嵌入式微处理器，能够同时有效支持控制密集型的应用和运算密集型的应用，具有广泛的应用前景。

参考文献

1 BDTI, Microprocessors vs. DSPs: Fundamentals and Distinctions, 2004
2 Patterson D A, Hennessy J L. Computer Architecture: A Quantitative Approach. 2nd ed. San Francisco: Morgan Kaufman Publish, 1996
3 张晨曦，王志英，等. 计算机体系结构. 高等教育出版社，2000
4 Cuadrado D L, et al. A platform-based comparison between a digital signal processor and a general-purpose processor from an embedded systems perspective. Embedded Systems Group, CPK, Aalborg University, 2002
5 Frederisen A, Christiansen R, Bier J, Koch P. An Evaluation of Compiler-Processor Interaction for DSP Applications. In: Proc. 34th IEEE Asilomar Conf. on Signals, Systems and Computers, Pacific Grove, CA, USA, Oct. 2000
6 任丽香，马淑芬，李方慧. TMS320C6000系列 DSP 处理器的原理和应用. 电子工业出版社，2000
7 钟文政，柯鸿禧. DSP TMS320C50原理与应用. 中国水利水电出版社，2003
8 Frantz G. Digital signal processor trends, Texas Instruments, IEEE Micro, 2000, 20(6): 52～59
9 Eyre J, Bier J. The Evolution of DSP Processors, BDTI a BDTI White Paper, 2000. 1～9
10 http://www.DSPsolution.com/html.intro/intro_dsp.htm
11 http://www.embed.com.cn/forum/show.asp? boareID = 12&announceID=3147