基于开发者行为分析的Web资源推荐

杨君雯 王 海 彭 鑫 赵文耘

(复旦大学软件学院 上海 201203) (上海市数据科学重点实验室(复旦大学) 上海 201203)

摘 要 现代的软件开发集成开发环境(IDE)为开发者提供了错误提示、代码补全、代码分析、版本管理等多方面的辅助开发支持,大大提高了开发效率。同时,开发者在日常开发过程中还常常依赖于互联网获取代码样例、配置说明、错误处理等Web开发资源。由于需要频繁地在IDE和浏览器之间进行切换并通过各种方式进行信息检索,开发者往往需要在Web开发资源的获取上花费大量的时间和精力。为此,提出一种基于开发者开发行为分析和挖掘的Web信息资源推荐方法。该方法通过自动记录和抓取开发者在IDE中的代码浏览和修改等动作以及在浏览器中的页面浏览信息获取基础信息。在此基础上,该方法从所抓取的浏览器页面中抽取结构化的信息资源,并通过聚类和基于时间的关联分析确定IDE开发行为与Web信息资源之间的相关性,从而在开发者在IDE中执行开发任务时自动推荐相关的Web信息资源。最后通过一个实验分析初步验证了所提方法的有效性。

关键词 Web资源,推荐,集成开发环境,行为监控,Web信息抽取

中图法分类号 TP311.5 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2017.07.027

Web Resource Recommendation Based on Analysis of Developer's Behavior

YANG Jun-wen WANG Hai PENG Xin ZHAO Wen-yun

(Software School, Fudan University, Shanghai 201203, China)

(Shanghai Key Laboratory of Data Science, Fudan University, Shanghai 201203, China)

Abstract Modern integrated development environment (IDE) provides developers with a variety of tools, including error warning, code complementary, code analysis, version control management, etc., to support software development and improve the developers' efficiency. However, such tools are deficient, as much more information, such as code sample, configure manifest, and error handling, is needed during development, and frequently switching between Web browser and IDE costs time and effort. A Web information resource recommendation method was proposed, which is based on the analysis of developer's behavior. The method extracts structured information including code samples from the developers' browsing history, and classifies them through text clustering. At the same time, the developer's behavior in the IDE was recorded. The relationship between WEB resources and developer's behavior will be established so that similar information can be recommended when the same situation happens. At last, an experiments was conducted to demonstrate that our method can save developing time efficiently.

Keywords Web resource, Recommendation, IDE, Behavior monitoring, Web information extraction

1 引言

现代的集成开发环境(IDE)为开发者提供了丰富的软件开发工具支持[1]。然而,面对纷繁复杂的软件技术,开发者仅依靠IDE往往不能进行高效的开发,还需要使用Web浏览器去搜寻Web资源。但是开发者可能并未对能够解决问题的Web资源进行记录,在下一次出现相同或者类似问题的时候,又会求助于相同的网站,从而花费大量的时间和精力在Web资源搜索和过滤上,大大降低了开发效率。

如果在开发过程中IDE能根据开发者的行为和正在开发的内容给予适当的推荐,将能够有效提高开发者的开发效率。想要达到这一目的,需要完整且准确地记录开发者在IDE中的开发行为和对Web资源的访问情况,并在两者之间建立起有效关联。在开发者遇到相似问题时,根据已有的关联就能给出相关推荐。

本文提出了一种基于开发者开发行为分析和挖掘的Web信息资源推荐方法,并根据该方法实现了一个原型工具。为了验证所提方法和实施工具的有效性,进行了实验验证,对比两个开发者先后的开发任务中的Web资源浏览记录以及任务完成时长,实验结果表明,本文所提方法能够对

到稿日期:2015-11-30 返修日期:2016-03-02 本文受国家自然科学基金(61370079),国家高技术研究发展计划(863)(2013AA01A605)资助。

杨君雯(1994—),女,硕士生,主要研究方向为软件维护,E-mail:jwyang11@fudan.edu.cn;王 海(1991—),男,硕士生,主要研究方向为软件维护,E-mail:13212010019@fudan.edu.cn;彭 鑫(1979—),男,副教授,主要研究方向为软件维护与演化、软件产品线、自适应软件、移动计算与云计算等,E-mail:pengxin@fudan.edu.cn;赵文耘(1964—),男,教授,博士生导师,主要研究方向为软件工程、软件开发工具及其环境、企业应用集成(EAI),E-mail:wyzhao@fudan.edu.cn。

Web 资源进行有效推荐并提高开发者的开发效率。

本文第 2 节介绍相关工作；第 3 节介绍 Web 资源的推荐方法；第 4 节介绍本文实现的工具；第 5 节通过实验验证了本文方法的有效性；第 6 节总结全文并对相关问题进行了讨论。

2 相关工作

2.1 开发者开发行为的监控

有关开发者行为的研究，国内外已经有不少的相关工作和文献，如 Eclipse 中的 Mylyn 插件就是一种集成在 Eclipse 中的插件，其可以匿名记录开发者的相关行为和上下文并以 XML 的形式存储[2]。夏威夷大学开发的 Hackystat[3] 工具采用“传感器-服务器”模式来收集和分析软件开发过程中开发者产生的行为，这些行为在经处理之后会被上传到云端服务器。

2.2 Web 信息提取的技术

随着人们对 Web 信息提取技术越来越重视，提取技术随着需求的增加而不断丰富。近年来国内外涌现了多种信息提取方式，根据抽取原理和抽取方式的不同可以分为以下几类：基于自然语言的处理方式，是将网页文档视为文本来处理的；基于包装器归纳方式，使用已经定义好的信息抽取规则，将网络爬虫搜集到的 Web 页面的数据抽取出来并转换为特定格式的描述信息；基于本体方式，本体能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，明确定义这些词汇（术语）之间的相互关系；基于 HMTL 结构方式，用解析器将 Web 文档解析成语法树，通过半自动或者自动的方式产生抽取规则，然后把信息抽取转化为对语法树的操作来达到信息提取的目的；基于视觉的信息提取，其重要特征是能够处理网页上的特殊信息，视觉特征在发现和提取网页信息时起着十分重要的作用，通过视觉特性把网页分成不同的块，从而达到抽取信息的目的。

2.3 文本聚类分析

传统的基于文本内容的文本聚类将文本表示为文本模型，如 VSM(Vector Space Model)[5] 模型、N-gram 模型、基于短语的模型、基于概念的模型、文本的图表示及概率模型。文本特征抽取与权重计算的方法主要有 TF-IDF[6] 函数、布尔函数、频度函数、互信息、期望交叉熵、二次信息熵、信息增益等。然后应用标准的聚类算法（如 K-means[7] 算法、谱聚类[4] 等）对文本进行聚类。

3 基于开发行为的 Web 资源推荐

3.1 总体框架

本文提出的基于开发者行为分析的 Web 资源推荐方法的框架如图 1 所示。

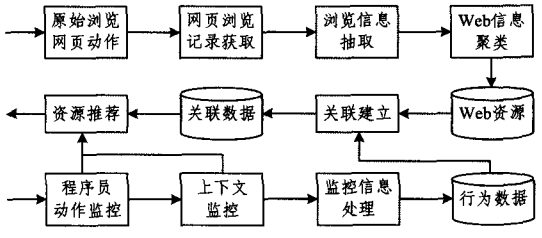

图 1 基于开发者行为分析的 Web 资源推荐方法的框架

方法的核心是基于 IDE 的行为监控和 Web 资源处理及资源推荐。当开发者使用 IDE 进行开发时，其进行的操作和相关上下文环境将被捕获并得以处理和保存。当开发者求助于网络，使用浏览器访问 Web 资源时，浏览器插件会获取用户 Web 浏览器的浏览历史，并抽取其中的信息；接着，系统对被抽取出来的 Web 信息进行文本聚类分析并保存，根据时间关系在编程行为和 Web 资源之间建立关联。在开发者进行相似操作时，相关的 Web 资源被推荐给开发者。

3.2 IDE 行为监控

IDE 行为监控是以开发者动作监控和上下文监控为起点的，它们都是随着后台的运行而运行，不间断地监听开发者所产生的操作及其所处的上下文信息。开发者动作监控主要是监控开发者当前动作的属性，上下文监控主要是监控开发者当前操作所处的 IDE 工作环境。

3.3 Web 资源捕获

3.3.1 基于启发式规则的网页信息抽取

在处理 Web 资源的过程中，首先要过滤掉 Web 资源中的无用信息，如网页中的一些导航栏、广告栏等。在过滤这些信息时使用启发式规则来判定哪些信息是无效的，这些规则包括：

1)〈script〉、〈noscript〉、〈!…〉、〈style〉等标签信息的内容；

2)display 属性为 none 或者高度太小的块（经过实验最后选取高度值为 10）；

3)尺寸不足 3×3（即三行三列）的表格。

对于有效信息，要确定信息的类型。表 1 列出了基于启发式规则的权值矩阵和详细解释。矩阵设置规则是根据经验分析得出的。

表 1 启发式规则矩阵详解

分类	对应规则	权值
日期	包含“时间”、“发布时间”、“提问时间”等	0.5
	出现特定的日期时间格式	0.5
来源	含有“来源”、“来自”、“转自”、“提问者”，或者文本节点为“xx 网”	0.6
	前一个或后一个为发布时间节点	0.2
	前一个或后一个为发布标题节点	0.2
正文	长度大于一个值	0.3
	多个段落组成或者多个相似标签组成	0.25
	包含多个换行符	0.25
	位于发布时间下方	0.2
标题	长度在一个范围内	0.2
	包含粗体字〈strong〉〈b〉标签	0.2
	包含 h1,h2,h3 标签	0.35
	满足上述 3 个条件，第一个为标题	0.25

3.3.2 Web 信息聚类分析

本文中采用的是基于 Lingo 算法[8] 的聚类分析。在进行网页聚类时，需要保证结果簇的内容和标签对用户是有意义的。Lingo 选择有意义的描述并且基于这些描述来选择内容，这些描述要满足两个条件[9]：相互差异大；尽可能覆盖到所有的输入集。下面是所使用的基于 Lingo 进行聚类的步骤。

1)预处理。语言识别，识别输入集合的语种；结束词标记，用于过滤掉以结束词结尾的短语。

2)特征提取。只有具有以下特性的短语才可能成为一个聚类标签：至少出现特定次数；没有越过句子边界；完整的短语；不以结束词开头或者结尾的句子。

3）聚类标签归纳。建立术语文档矩阵，与 SHOC[12] 不同的是，Lingo 采用单独的术语进行矩阵构建，因为多数情况下是使用单个术语去表述一个抽象概念（因为单独的术语能够获得更小粒度的描述）；挖掘抽象概念，在这一步中使用 Singular Value Decomposition[11] 来获取列空间的正交基地；阈值选取，选取能继续抽象的概念供下一步使用；短语匹配；优化候选标签。

4）聚类内容挖掘。使用 Vector Space Model[10] 把输入片段赋值给上一步产生的聚类标签。

5）最终的聚类信息得分通过式（1）进行计算。label-score 是通过聚类标签归纳中的矩阵来计算的。

$$\text{cluster-score} = \text{label-score} \times \text{member-count} \quad (1)$$

3.4 资源关联及推荐

3.4.1 关联建立

通过时间段匹配的方式对 IDE 操作行为和 Web 资源建立关联关系，并作为资源推荐的基础。每一个 Web 资源都和相应的 IDE 操作中的某个上下文中的文档之间建立直接联系，每一个上下文同时又和自己所包含的引用类建立联系，并且每一个 Web 资源也与聚类分析之后自己隶属的簇存在直接联系。

开发者的每个 IDE 操作以及每个浏览器的浏览行为都有开始和结束的时间戳信息，这些信息使得开发者的所有行为被顺序地置于时间轴上。系统根据相应的时间信息进行分析。当一个网页的打开时间正好处于一个项目文档的阅读时间之中时，我们则认为这两者之间是存在关联的，就将上一步所记录下来的某个文档打开和结束阅读的两个时刻作为衡量标准，凡是浏览时间正好在两个时刻之间的网页，都将和这个文档相关联。

3.4.2 资源推荐

通过对 IDE 中用户的上下文监控，可以捕获用户上下文信息。如果上下文中的信息出现在已有的关联关系中，系统将对其所关联的 Web 资源进行推荐。

4 工具实现

根据第 3 节中提出的方法，本节实现了一个原型工具。

4.1 工具设计

IDE 监控插件工具是在 Eclipse Luna Service Release 2 (4.4.2)版本中进行开发的。在设计之前，先行确定本插件应达到的技术要求。

1）完整性：插件的生命周期和 Eclipse 保持一致，插件应当完整地记录从打开到关闭 Eclipse 的所有行为。

2）自动化：插件的启动、记录、退出等操作应当由插件本身和其所处的环境来决定，不需要用户主动干预。

3）无干扰性：插件对于开发者行为的监控应该在后台进行，做到最小程度地影响开发者的开发行为。

4.2 IDE 监控的实现

4.2.1 IDE 动作监控

Eclipse UI 行为主要指开发者和 IDE 界面的各种 UI 元素进行的交互事件。在我们的工具中，监听的是用户的选择事件和页面事件。选择事件在用户选择了一个项目或者一个文件时被触发。Eclipse 的 Properties 视图将会显示与资源相关的属性，当选择了某个方法或属性时，大纲视图将会定位到相关的位置。页面事件是页面在打开、激活和关闭时所触发的事件。页面位于工具栏和状态栏之间，由若干个组件(Part)组成。页面事件监控接口由 org. eclipse. ui 包提供。

4.2.2 上下文监控

从开发者打开 IDE 的那一刻起，每一个行为几乎都是施加于某一个工作空间的对象上，它可以是一个项目、一个包、一个文件等。为了更加客观地监控开发者行为，需要对其操作所施加的对象进行分析，这种信息可以为后续的开发者的开发行为分析提供丰富且准确的信息。结合时间信息，还可以理清用户开发过程中的思路等信息。本文工具考虑的是基于结构的选择事件的上下文信息。当开发者通过鼠标、键盘等对视图中的结构性对象进行选择时，Eclipse 会产生一个 StructuredSelection 的实例。结构化选择项广泛存在于 Eclipse环境中，如包资源管理器中的包和文件等。

4.3 信息推荐的实现

如果一个文件出现在关联数据库中，那么就从关系数据库中将编程人员正在处理的任务的相关 Web 资源推荐给编程人员。首先根据 priority 将排行前 3 的抽取结果在 IDE 里展示出来，如果用户认为这 3 条信息不能够解决自己的问题，可以根据需求继续展开剩下的推荐信息。同时，对每一条信息，我们会根据聚类分析得到的结果把与该条信息处于同一类的资源也推荐给用户。

5 实验

为了验证本文所提出的基于 IDE 开发行为的 Web 资源推荐方法和实现的工具的有效性，把该方法和原型工具运用到开发者真实的开发动作的捕获中。通过案例实验，本文希望验证如下问题：

1）本文的方法是否能够在开发者进行相似开发行为时提供有效的推荐？

2）本文的方法是否能够有效提高开发者的开发效率？

5.1 实验设计

设计了两组任务，每组任务包含两个相似的任务：

任务 1.1　读取 history. xml 文件并存入数据库中；

任务 1.2　读取 record. xml 文件并转换时间格式；

任务 2.1　实现 Dijkstra 最短路算法；

任务 2.2　实现 Floyd-Wallshall 最短路算法。

其中任务 1.1 和任务 2.1 被分别分配给两位开发者 A 和 B，在他们完成各自任务后，双方交换分组，以减小开发能力差异，由 A 完成任务 2.2，B 完成任务 1.2。最后，查看在任务 1.1 和任务 2.1 过程中建立的资源关联和 4 个任务被完成的时间。另外，要求两位开发者分别标注被采用的 Web 资源。安排开发者 C 和 D 组成对照组，在没有推荐系统的情况下完成相同的任务，即 C 完成任务 1.1 和任务 2.2，D 完成任务 1.2 和任务2.1。

5.2 实验结果

5.2.1 有效推荐

表 2 列出了在完成任务 1.2 和任务 2.2 的过程中系统给出的 Web 资源的推荐次数。在任务 1.2 中，开发者一共查看了 12 次 Web 资源，其中 2 次来自于系统的推荐，并且在前两

个推荐资源中就找到了正确的资源。在任务 2.2 中，开发者一共查看了 35 次 Web 资源，其中 7 次来自系统推荐。

表 2 Web 资源推荐数量

任务	查看 Web 资源的次数	成功推荐次数	推荐资源位置
任务 1.2	12	2(16.7%)	1—2
任务 2.2	35	7(20%)	1—3

在任务完成后，通过对开发者 A 和 B 的实验过程复述的总结可知，开发者 B 在完成任务 1.2 时，在读取 XML 文件的过程中，对本系统推荐资源排名第一但不是 java 编写的源代码选择了忽略，而是选择了排名第二的 java 代码，且仅仅修改了对应的属性名称和路径名称就成功使用了推荐资源中的代码。在进行 String 和 Date 的转换时，开发者 B 从排名第一的推荐资源中找到了解决方案，而其浏览其他非推荐资源的目的在于寻找更好的解决方案(例如使用 java 标准库中的方法)。

开发者 A 在完成任务 2.2 时，首先需要了解任务所设计的算法。在这个过程中，我们的系统推荐的资源给了 A 良好的提示。在实现 Floyd-Wallshall 算法时，第一条推荐是用 C 语言实现的代码，A 虽然并没有直接采用其中的代码，但通过阅读代码很快掌握了算法的基本流程。第二条推荐资源是一个 ZOJ 的原题，A 通过该题目在网络上找到了解决问题的源代码。然而，A 在实现过程中由于写错了循环顺序，需要求助其他 Web 资源来解决 bug，由于该 debug 过程并未出现在任务 2.1 中，因此系统并不能给出有效推荐。

从以上数据和完成任务的过程可以看出，在已经发生过类似开发行为的情况下，我们的系统能够给出正确的资源推荐，辅助开发者完成开发任务。

5.2.2 效率提升

表 3 列出了 4 位开发者完成任务所花费的时间。从对照组可以看出，任务 1.1 和任务 1.2、任务 2.1 和任务 2.2 的完成时间分别相近，这表明每组任务中的两个任务的难度相近，即它们为相似任务。而实验组中，在使用该系统进行推荐之后，完成时间明显缩短，表明在有推荐的情况下，开发时间大大缩短了，开发者的效率得到了提高。

表 3 任务完成时间

开发者	实验组		对照组	
	A	B	C	D
任务 1.1	30min49s	/	40min55s	/
任务 2.1	/	42min37s	/	45min2s
任务 1.2	/	11min35s	/	1h8min
任务 2.2	24min35s	/	50min14s	/

6 总结和讨论

本文提出和实现的工具仍有一定的局限性：对开发者开发行为的动作监控和上下文监控的部分只是停留在了粗粒度的文本层面上；在对 Web 资源的提取上，也不能完全保证提取主体内容的完备性，对于结构特殊的 Web 资源可能会存在抽取内容和实际内容不相符的情况；同时，本文所提方法中对文本的聚类方式只是采用了 Lingo 这一种算法，并未对其他文本聚类算法进行尝试；虽然 Web 资源和开发者行为建立了关联，但是这种关联也只是基于时间段的，并没有在语义上进行相关的联系。我们还可以更多地将这些数据及时转变成更有价值的信息，结合语义信息进行高层泛化的抽象和可视化展示也将成为我们未来的优化方向。

结束语 本文提出了一个基于 IDE 行为监控的 Web 资源推荐方法。根据这个方法，本文实现了一个原型工具，使用 Eclipse 插件机制开发出能够对开发者行为进行监控的插件，并使用启发式规则对 Web 资源进行信息提取，同时使用 Carrot2 的 Lingo 算法对抽取出来的 Web 资源进行聚类分析，用于推荐给开发者。通过实验验证了本文实现的推荐方法可以给开发者提供有效的 Web 资源推荐从而提高开发者的开发效率。

参 考 文 献

[1] AMOR J J, ROBLES G, GONZALEZ-BARAHONA J M. Effort estimation by characterizing developer activity[C]// International Workshop on Economics Driven Software Engineering Research. ACM, 2006: 3-6.

[2] LAYMAN L, WILLIAMS L, AMANT R S. Toward reducing fault fix time: Understanding developer behavior for the design of automated fault detection tools[C]// Empirical Software Engineering and Measurement. IEEE, 2007: 176-185.

[3] NUYUN Z, GANG H, YING Z, et al. Automating Reusable-Procedure Discovery through Developer's Action Analysis [C]// 2010 10th International Conference on Quality Software (QSIC). IEEE, 2010: 240-247.

[4] JIN X, ZHOU Y, MOBASHER B. A maximum entropy Web recommendation system: combining collaborative and content features[C]// ACM SIGKDD International Conference on Knowledge Discovery in Data Mining. 2005: 612-617.

[5] WEI C, SEN W, YUAN Z, et al. Algorithm of mining sequential patterns for web personalization services [J]. ACM SIGMIS Database, 2009, 40(2): 57-66.

[6] BRODER A Z, GLASSMAN S C, MANASSE M S, et al. Syntactic clustering of the Web[J]. Computer Networks and ISDN Systems, 1997, 29(8): 1157-1166.

[7] HOBBS J R, APPELT D, BEAR J, et al. 13 FASTUS: A Cascaded Finite-State Transducer for Extracting Information from Natural-Language Text[J]. Finite-state Language Processing, arXiv: cmp-lg/9705013V1, 1997: 383.

[8] CHANG C H, LUI S C. IEPAD: information extraction based on pattern discovery[C]// International Conference on World Wide Web. ACM, 2001: 681-688.

[9] CHANG M L, LIN Y C, GUO L F. Design and implementation of an efficient Web cluster with content-based request distribution and file caching[J]. Journal of Systems and Software, 2008, 81(11): 2044-2058.

[10] SHAHABI C, BANAEI-KASHANI F, CHEN Y S, et al. Yoda: An accurate and scalable Web-based recommendation system [M]// Cooperative Information Systems. Springer Berlin Heidelberg, 2001: 418-432.

[11] OSIŃSKI S. An algorithm for clustering of Web search results [D]. Poznań University of Technology, Poland, 2003.

[12] WANG F H, SHAO H M. Effective personalized recommendation based on time-framed navigation clustering and association mining[J]. Expert Systems with Applications, 2004, 27(3): 365-377.