

引用格式:张国生. 面向特征的软件需求规约[J]. 微电子学与计算机, 2019, 36(11): 20-23. [ZHANG G S. Feature-oriented software requirements specification [J]. Microelectronics & Computer, 2019, 36(11): 20-23]

面向特征的软件需求规约

张国生

(云南大学 信息学院, 云南 昆明 650500)

摘 要: 基于大数据方法对自然语言用户需求文本进行分析, 用 TF-IDF 方法计算特征词在需求文本中的出现频率, 归一化后求得特征词在文本中的权重, 用卡方统计量衡量特征词和簇之间的相关性, 用 k-均值算法和二分 k-均值算法对特征集聚类, 生成用户需求文本特征树, 并映射为需求功能分解树中对应的需求功能, 使需求特征语义、结构更加清晰, 增强了自动需求规约能力, 提高了需求分析效率, 使需求规约更加科学、合理、准确。

关键词: TF-IDF; 卡方; k-均值; 二分 k-均值; 特征树; 功能分解树

中图分类号: TP311

文献标识码: A

文章编号: 1000—7180(2019)11—0020—04

Feature-oriented software requirements specification

ZHANG Guo-sheng

(School of Information, Yunnan University, Kunming 650500, China)

Abstract: Based on big data method, the text of natural language user requirement is analyzed. TF-IDF method is used to calculate the frequency of feature words appearing in the requirement text. After normalization, the weight of feature words in the text is obtained. Chi-square statistics is used to measure the correlation between feature words and clusters. The feature set is clustered by k-means algorithm and bisecting k-means algorithm, and the user requirement text feature tree is generated, which is mapped to the corresponding requirement function in the requirement function decomposition tree. The semantics and structure of requirement feature are clearer. The ability of automatic requirement specification is enhanced. The efficiency of requirement analysis is improved, and the requirement specification is more scientific, reasonable and accurate.

Key words: TF-IDF; Chi-square; k-means; bisecting k-means; feature tree; function decomposition tree

1 引言

特征是用户对系统的感知, 用特征对需求规约进行模块化组织是一种非常自然的手段^[1]. 特征是不同的涉众之间有效沟通的媒介, 客户与开发人员通过特征共同讨论软件产品拥有或交付的功能. 特征具有明显的可识别功能, 是客户和开发人员都能理解的基本抽象.

特征是一组逻辑相关的功能需求集合, 为用户提供了满足特定业务需求的能力^[2]. 特征是贯穿软

件生命周期和跨越问题空间和解空间的一阶实体, 特征的引入能够减少用户和软件开发者之间的期望差异. 特征是软件需求规约中一组相互紧密联系的需求构成的模块^[3]. 从规约需求的角度看, 特征是影响终端用户的系统属性, 提供了一种分割、组织需求的模式, 将特征看作需求空间的一阶实体, 系统具有的诸多特征及其相互之间的关系共同构成软件系统的需求空间. 从需求内涵的角度看, 特征反映了软件系统具有的能力和特点, 体现了需求抽取的涉众对系统的期望和理解; 从需求类型的角度看, 特征可以

表现为用户对系统的功能性需求、非功能性需求及其系统外部环境对系统的约束。

面向特征规约需求的方法,是将多组相对稳定的特征及其相互之间的依赖关系,从不同的侧面表征软件系统的需求。而传统的规约需求的方法,常用自然语言对系统需求进行描述,辅之以图形、可视化界面增强涉众对需求的分析和理解,没有有效地表征需求与需求之间的结构关系和语义关系,仅仅是将软件需求简单地罗列出来。

2 特征分析和特征依赖

2.1 特征分析

特征是系统具有的显著、有特色的品质、属性、方面和特点,也是特定领域中涉众捕获不同软件系统的共性及其变化性的有效基本单元。在领域需求分析中所用的特征模型,既蕴含了特定领域的复用特征及其特征间的关系,也为系统功能模型的参数化提供了依据。

功能性特征是应用系统提供的基本服务。功能分析的主要作用是识别并建模领域中应用系统具有的共性和变化性功能,并以结构化的方式组织,通过实例化或定制,得到适合具体应用系统的功能需求模型。

功能规约是通过软件系统内部数据、活动、行为、输入、输出,以及它们之间的逻辑关系和数据流关系描述系统的结构。功能模型按照不同的层次,对系统功能进行全面描述,功能分解帮助分析人员理解问题。一般/特殊的概念帮助定义具有一般性的功能或对象,应用系统的规约可以看成一般性功能和对象的特殊化。

2.2 特征依赖

特征依赖是特征与特征之间的关系,可以是特征内在的层次结构关系,也可以是业务逻辑的约束关系,或者运行依赖关系等,由于用户需求变更造成变体选择的不确定性,增加了特征依赖的复杂度,因此,识别、描述特征依赖是特征建模的重要课题。特征之间的依赖关系有四种:精化、交互、影响和约束关系。

精化关系是存在于不同粒度或抽象层次的特征间的关系,这些特征通过精化形成层次结构,层次结构提供了一种描述复杂系统的有效手段,从抽象到具体,从高层到低层,按照层次结构去理解、把握相对复杂的系统。精化关系分成三种类型:分解、属性化和特殊化。把一个特征精化为一组作为其构成成

分的子特征称为分解,分解体现了整体特征与部分特征之间的控制与协调,也体现了整体特征与部分特征在逻辑上的紧密关联。识别出一个特征具有的属性称为属性化。把一个特征精化为包含更多细节的特征称为特殊化,常用于描述一个特征包含的一组可能的变体。

交互关系是软件系统组成的关键要素,交互是特征间的动态依赖关系,是约束在软件系统运行时的体现,交互将各个相对独立的成分组装成一个有机的整体。

影响关系刻画了特征在软件规约层次上的静态依赖关系,从特定的角度描述责任的分配,有助于开发者分析系统需求的变化对软件设计、实现的影响。约束关系是不同特征的绑定状态之间的依赖关系,可以验证特征模型剪裁结果的完整性、一致性。

特征模型可以用树形结构表示,特征树能较好地表征特征之间内在的层次结构关系和依赖关系。

3 向量空间模型

聚类是将数据集按照元素间的相关性划分为若干簇的过程,其目的是将文本集划分为若干簇,使同一个簇内文本间的相似度最大,不同簇之间文本的相似度最小。

向量空间模型 VSM 是由 G. Salton 于 1975 年提出的^[4],其基本思想是:给定一自然语言文档 $d = d(t_1, w_1; t_2, w_2; \cdots; t_N, w_n)$, 其中, t_i 是从文档 d 中选出的特征词, w_i 是 t_i 的权重, $1 \leq i \leq n$, 将 $t_1, t_2, \cdots, t_n$ 看成一个 n 维坐标系,而 $w_1, w_2, \cdots, w_n$ 为相应的坐标值,则 $d(w_1, w_2, \cdots, w_n)$ 被看成 n 维空间中的一个向量。

3.1 TF-IDF

TF-IDF 是一种统计方法,用于评估某个词对于一个文本或一个文本集的重要程度,是信息检索和文本挖掘的常用统计加权技术。

一个词的重要程度与其在文本中出现的频率成正比关系,而与其在所有文本集中出现的频率成反比关系。TF-IDF 对词的加权计算,综合了这两方面的因素。设 d 是文本集中的文本, f 是文本集的特征词,则

(1) 词频 $TF(f, d)$ 为特征词在文本中出现的频率,表征词对该文本的重要程度。设 f 为特征词, f 在文本 d 中出现的次数为 m_{fd} , z 是 d 中所有的特征词,则

$$TF(f, d) = \frac{m_{fd}}{\max_z m_{zd}}$$

(2) 倒排词频 IDF_f 为特征词在文本集中出现的频繁程度,用于削弱那些在文本集中过于频繁出现的词的重要程度,因为这些词常常对文本没有显著的区分能力。

$$IDF_f = \log_2(N/(n_f + 0.01))$$

式中, N 为文本集中的文本总数; n_f 为出现特征词 f 的文本数。

$TF(f, d) * IDF_f$ 为特征词 f 在文本 d 中出现的频率,可以滤除文本中的常见词,保留重要词,对其归一化,则特征词 f 在文本 d 中的权重为

$$W(f, d) = \frac{TF(f, d) * IDF_f}{\sqrt{\sum_{f \in d} [TF(f, d) * IDF_f]^2}}$$

3.2 卡方统计量

卡方统计量用于衡量特征和簇之间的相关性,对于簇 c ,特征 f 的卡方统计量为

$$\chi^2(f, c) = \frac{N(AD-BC)^2}{(A+C)(B+D)(A+B)(C+D)}$$

式中, N 为文本集中的文本总数; A 为特征 f 在簇 c 中出现的频数; B 为特征 f 在除簇 c 以外的其它簇中出现的频数; C 为除特征 f 以外的其它特征在簇 c 中出现的频数; D 为除特征 f 以外的其它特征在除簇 c 以外的其它簇中出现的频数; $N = A + B + C + D$ 。

设簇集 $C = \{c_1, c_2, \dots, c_k\}$, 特征在文本集中的重要程度用以下两种方法表征。

(1) 特征在不同簇中的卡方统计量的平均值

$$\chi_{\text{avg}}^2 = \frac{1}{k} \sum_{c \in C} \chi^2(f, c)$$

(2) 特征在不同簇中的卡方统计量的最大值

$$\chi_{\text{max}}^2 = \max_{c \in C} \{\chi^2(f, c)\}$$

4 k-均值算法和二分 k-均值算法

4.1 k-均值算法

k-均值算法由 MacQueen 提出^[5],是聚类划分的一种经典方法,基于质心聚类,算法因原理简单,收敛速度快,适应性强而被广泛应用。其基本原理是:随机选取 k 个数据点作为初始聚类中心,每个数据点为一个簇的质心,通过最小化簇内误差平方和 SSE,将其分配到最近的簇,重新计算每一个簇的平均值,作为下次聚类新的质心,不断迭代改变聚类质心,从而改变数据划分,直到评估函数 SSE 收敛。

$$SSE = \sum_{i=1}^k \sum_{x \in c_i} \text{dist}(m_i, x)^2$$

式中, m_i 为簇 c_i 的聚类中心; x 为该簇中的一个样本; $\text{dist}(m_i, x)$ 为聚类中心 m_i 与样本 x 之间的欧氏距离。

由于初始的 k 个聚类中心是随机选取的,没有考虑数据的实际分布,易收敛于局部最优解,导致聚类结果不准确。

4.2 二分 k-均值算法

二分 k-均值算法是对 k-均值聚类算法的改进和扩充^[6],其主要思想是:假设要将数据分为 k 个簇,先用 k-均值算法将所有数据分成两个簇,从所得结果中选择较大的簇,对其使用 k-均值算法进行分裂操作,不断选择较大的簇用 k-均值算法进行分裂操作,直到分成 k 个簇为止。

二分 k-均值算法如下:

输入:数据集 D ,二分次数 m ,簇数 k 。

输出:簇集 $C = \{c_1, c_2, \dots, c_k\}$ 。

(1) 初始簇集 C 为包含所有数据的簇,将生成的簇数 k' 初始化为 1。

(2) 从簇集 C 中取出一个最大的簇 c 。

(3) 用 k -均值聚类算法对簇 c 进行 m 次二分聚类操作,得到 m 对子簇。

(4) 分别计算这 m 对子簇的总 SSE 的大小,将最小总 SSE 的一对子簇添加到簇集 C 中, $k'++$ 。

(5) 如果 $k' = k$,结束算法;否则重复步骤(2)~(5)。

二分 k -均值算法不需要初始化,其每一步操作就是从 m 对子簇中找出误差平方和最小的一对子簇,然后再进行 k -均值聚类。

5 面向特征的需求规约

5.1 软件需求特征树

需求工程将需求分为三个不同的层次:功能需求、用户需求和业务需求。功能需求规定开发人员在软件中实现的功能,利用这些功能完成任务,满足业务需求。用户需求描述了用户的目标或者用户要求系统完成的任务。业务需求表示组织或用户高层次的目标,描述了组织为什么要开发系统,即希望达到的目标。

系统的需求开发是为了获得整个系统的期望目标,包括功能特征和非功能特征,需要判断软件系统的涉众,采集他们的期望、目标与要求,研究系统环境,确定系统约束,对软件进行整体需求分析。

用户对软件提出的需求和期望,一般都以文本的形式用自然语言记录下来,所有的软件需求形成

一个文本集合,设有一自然语言文本集 D , 包含 m 个文本 $D = \{d_1, \dots, d_m\}$, 文本 d 由 n 个词组成 $d = \{w_1, \dots, w_n\}$. 用 Python 语言的 jieba 对文本集进行分词, 滤除停用词、无意义词等, 保留所有特征词, 计算特征词的词频 $TF(f, d)$, 作为特征词在文本中出现的频率, 计算倒排词频 IDF_f , 用 $TF(f, d) * IDF_f$ 作为特征词 f 在文本 d 中出现的频率, 归一化后求得特征词 f 在文本 d 中的权重 $W(f, d)$, 滤除文本中的常见词, 保留重要词, 计算卡方统计量 $\chi^2(f, c)$, 用于衡量特征和簇之间的相关性, 得到特征集 F , 用 k -均值算法与二分 k -均值算法对 F 聚类, 得到相应的簇集 $C = \{c_1, c_2, \dots, c_k\}$, 分别取前 m 个特征值对应的特征, 生成软件系统的用户需求特征树。

5.2 软件需求功能分解树

需求工程过程从最初的用户需求或者更高一级的面向系统的需求开始, 逐步分解并完善这些需求, 直到满足整个软件系统的需求为止^[7]. 从软件功能实现的角度, 结构化方法将整个软件系统自顶向下分解, 一个功能被分解成多个子功能, 再对子功能继续分解, 直到不能再分或者不需要再分为止, 系统功能和子功能之间的关系组织成层次结构, 一个具体软件的功能需求, 可以用分层的方法来组织和管理, 分层将一个复杂的问题分解成多个子问题, 对子问题进一步精化, 又可以分解成更小的子问题, 直到得出需求的细节层次, 这些功能的层次集合构成一棵软件需求功能分解树, 用基本需求功能及其组合实现需求功能分解树中的树叶结点, 解构出整个软件系统满足涉众要求的功能需求模型。

从涉众需求文本集合中提取的用户需求特征树, 描述了涉众对软件系统的目标和期望, 而需求功能分解树则从业务的角度, 表征了软件系统所具备的功能, 将用户需求特征树中的各层的特征集合, 映射为需求功能分解树中对应层的需求功能集合. 从自然语言书写的用户需求文本中规约出需求特征树中的层次特征集, 从而自然映射为软件需求层次功能, 提高了需求分析的效率, 避免了分析的随意性, 减少了需求分析的经验成分, 增加了需求分析的科学性、合理性, 提高了软件需求规约的准确率, 降低

了人力成本, 为数据驱动的自动软件工程奠定了基础。

6 结束语

基于大数据方法对自然语言用户需求文本进行分析, 用 TF-IDF 统计方法、 k -均值算法和二分 k -均值算法, 对自然语言需求文本进行特征规约, 从文本中抽取特征词, 求其权重, 生成需求文本特征树, 并映射为需求功能分解树中对应的需求功能, 使需求特征语义和结构更加清晰, 增强了自动需求分析能力。

参考文献:

- [1] DAVIS A M. The design of a family of application-oriented requirements languages[J]. Computer, 1982, 15(5):21-28.
- [2] WIEGERS K E. Software requirements[M]. Washington:Microsoft Press, 1999.
- [3] TURNER C R, FUGGETTA A, LAVAZZA L, et al. A conceptual basis for feature engineering[J]. Journal of Systems and Software, 1999, 49(1):3-15.
- [4] SALTON G, WONG A, YANG C S. A vector space model for automatic indexing [J]. Communications of the ACM, 1975, 18(11): 613-620.
- [5] MACQUEEN J. Some methods for classification and analysis of multivariate observations[C]//Proc. of the 5th Berkeley Symposium on Mathematical Statistics and Probability. Los Angeles: University of California Press, 1967:281-297.
- [6] SAVARESI S M, BOLEY D. On the performance of bisecting k-means and PDDP[C]//Proc. of the 1st SIAM International Conference on Data Mining. Chicago, USA:[s. n.], 2001:1-14.
- [7] 张国生. 需求演化过程建模[J]. 微电子学与计算机, 2012, 29(5):54-57.

作者简介:

张国生 男, (1968-), 硕士, 副教授. 研究方向为软件工程.
E-mail: zhanggs@ynu.edu.cn.