

# 基于语义匹配模型的软件测试用例共享和复用方法分析

刘阳, 寇力, 杨基宏, 吕燕, 祝宇, 王鹏钧  
(中国人民解放军96901部队, 北京 100094)

**摘要:** 阐述基于语义匹配模型软件测试用例共享与复用方法, 以提高软件测试效率和质量。通过构建软件测试知识库, 利用语义相似度从知识库中挖掘相似度最高的测试用例, 并根据不同测试项目的需求进行修改与定制, 实现测试用例的高效共享与复用。

**关键词:** 语义匹配模型, 软件测试用例, 相似度计算, 召回率, 准确度。

中图分类号: TP183, TP311.53

文章编号: 1000-0755(2025)02-0083-03

文献引用格式: 刘阳, 寇力, 杨基宏, 吕燕, 祝宇, 王鹏钧. 基于语义匹配模型软件测试用例共享和复用方法分析[J]. 电子技术, 2025, 54(02): 83-85.

## Analysis of Software Test Case Sharing and Reuse Methods Based on Semantic Matching Model

LIU Yang, KOU Li, YANG Jihong, LYU Yan, ZHU Yu, WANG Pengjun  
(PLA 96901, Beijing 100094, China.)

**Abstract** — This paper expounds the method of sharing and reusing software test cases based on semantic matching models to improve software testing efficiency and quality. It constructs a software testing knowledge base, uses semantic similarity to mine the test cases with the highest similarity from the knowledge base, and modifies and customizes them according to the needs of different testing projects, achieving efficient sharing and reuse of test cases.

**Index Terms** — semantic matching model, software test cases, similarity calculation, recall rate, accuracy.

### 0 引言

随着软件的迅速发展, 软件测试的应用范围越来越广, 不断重新开发软件测试用例会消耗过多的财力与人力<sup>[1]</sup>。然而测试用例共享和复用模块实现了测试用例的共享和复用, 测试用例修改和定制模块实现了对共享测试用例的修改和定制, 测试用例管理和版本控制模块实现了测试用例的管理和版本控制, 从而提高了软件测试效率和质量。

虽然大量高质量的可用测试用例已被应用于软件工程领域, 但复用技术仍没有很好的发展<sup>[2,3]</sup>。为此研究基于语义匹配模型软件测试用例共享和复用方法, 基于测试条件快速生成新的测试用例, 提高软件测试效率。

### 1 研究背景

基于语义匹配模型软件测试用例共享和复用方法研究。设计与编写软件测试用例需要许多人力、物力与时间, 可以通过用例共享与复用的方法节约时间, 提高软件测试效率。

软件测试用例是从测试用例库中挑选出符合用户需要的测试用例<sup>[4]</sup>。首先构建基于语义匹配模型软件测试用例共享和复用模型, 然后对于模型中

的三个部分以此进行介绍。

基于语义匹配模型软件测试用例复用模型。基于语义匹配模型软件测试用例复用模型具体流程如图1所示, 共包括3个核心部分。

(1) 构建软件测试知识模型: 构建软件测试知识模型是本文方法的重要前提, 其过程主要通过标注、抽取、融合以及构建知识库等过程将一个杂乱无章的原始软件测试用例语料整理成逻辑清晰、分类明确、能够进行反复复用的知识图谱, 提高软件测试用例知识使用与应用的便利性<sup>[5]</sup>。

(2) 基于语义相似度的语义匹配: 根据用户需要的测试用例的语义从测试用例库中匹配到关联度更高的软件测试用例类别, 从具体分类中再挑选出相似度更高的软件测试用例用于共享与复用。

(3) 测试用例相似度计算: 通过基于语义相似度的语义匹配后可以确定问题模板的具体分类, 然后再通过知识图谱获取用户需要的答案。

### 2 构建软件测试知识模型

#### 2.1 软件测试用例知识获取和应用框架结构

为更好实现软件测试用例复用, 需要对软件测试知识获取方法进行研究, 重点在于怎样挖掘出

**作者简介:** 刘阳, 中国人民解放军96901部队, 高级工程师, 博士研究生; 研究方向: 软件测试。  
收稿日期: 2024-01-10; 修回日期: 2025-02-12。

软件测试用例知识，首先通过语义分析以及信息抽取技术将测试报告与测试用例等非结构化自然语言资料转化为结构化的知识表现形式，并构建成可推理、有索引、可视化的软件测试知识库；基于深度学习的软件测试知识获取与应用方法的具体过程共分为四个步骤，具体如图2所示。

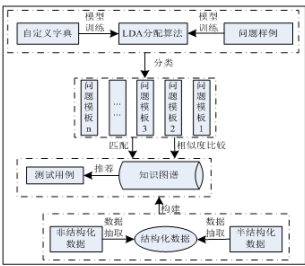


图1 基于语义匹配模型的软件测试用例复用模型具体流程

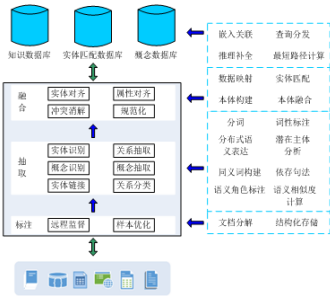


图2 基于深度学习的软件测试知识获取与应用方法

根据图2可见，基于深度学习的软件测试知识获取与应用方法的四个步骤分别是标注、抽取、融合以及构建数据库。通过文档分解、结构化存储完成软件测试知识标注；实现软件测试知识抽取则需要需要进行分词、词性标注、分布式语义表达、潜在主体分析、同义词构建、语义角色标注、语义相似度计算等多个步骤；完成软件测试知识的融合需要进行数据映射、实体匹配、本体构建与本体融合；最终通过嵌入关联、查询分发、推理补全以及最短路径计算完成软件测试知识数据库的构建。通过上述过程将软件测试知识挖掘出，为提高软件测试用例复用的匹配度，需将其转化为知识图谱，最终通过知识卡片的形式展示给用户。

2.2 构建知识图谱

构建知识图谱共包括特征提取、本体构建、生成图谱三个部分构成。

(1) 特征提取：本文通过项目测试文档中的数据构建知识图谱，而项目测试文档中的数据属于半结构化的数据，因此基于测试用例特征以及专家相关经验，从这些半结构化数据中提取出与软件测试用例有关的特征数据，并转化为结构化数据，软件测试用例具体设计结果如表1所示。

(2) 构建本体：本体属于知识图谱的模式层，其用于描述一整个领域的数据集。在软件测试用例的本体设计中通过七步法完成本体的构建。软件测试用例的核心本体包括问题报告单、项目、测试用例信息、测试项、测试项明细以及测试记录等数据，多方面的展现测试用例的关键信息。

(3) 图谱生成：软件测试用例知识图谱通过颜色将实体区分开，两个实体之间的连接即表示两者之间的关系。通过软件测试用例知识图谱能够简单明了的看清项目、问题报告单与测试用例之间的关系，并能发现测试用例的分布、缺陷的分布。在软件测试用例复用过程中，能够率先匹配到发掘软件缺陷的测试用例。

3 基于语义相似度的语义匹配

软件测试用例复用模型不仅需要搜索到用户所需要的测试用例，还需挖掘出与测试用例相关的信息。用户输入具有多属性内容的测试用例，然后基于知识图谱获取到这些属性的测试用例集，最终根据语义之间的相似度完成软件测试用例匹配，获取到与用户所需相似的软件测试用例，实现软件测试用例的共享与复用。

设定软件测试用例的集合为 $M = \{M_1, M_2, \dots, M_n\}$ ，软件测试用例属性为 $S = \{S_1, S_2, \dots, S_k\}$ ，属性相对应的权重为 $\omega = \{\omega_1, \omega_2, \dots, \omega_k\}$ 。将软件测试用例的所有属性进行分词并通过完成编码；因此 $t_{i,k}$ 代表软件测试用例 $M_i$ 在属性 $S_k$ 上编码为1的数量，那么可以将相似度计算转化为相似值计算，如式（1）。

$$N(M_r, M_i) = \sum_{j=1}^{i-k} t_{ij} w_j \tag{1}$$

式中， $N(M_r, M_i)$ 代表软件测试用例 $M_r$ （用户输入内容）与软件测试用例数据库集合中 $M_i$ 的相似值。然而 $N$ 的表达式为式（2）。

$$N = \max \{N(M_r, M_1), N(M_r, M_2), \dots, N(M_r, M_i)\} \tag{2}$$

代表软件测试用例集合 $M$ 中 $N(M_r, M_i)$ 的最大值。因此 $M_i$ 即为用户所需测试用例相似度最高的软件测试用例。

表1 软件测试用例具体设计结果

项目名称	***	用例标识	***
用例描述	测试系统初始化完成后。		
测试类型	<input type="checkbox"/> 静态分析 <input type="checkbox"/> 代码走查 <input type="checkbox"/> 逻辑测试 <input type="checkbox"/> 基本路径测试 <input type="checkbox"/> 功能测试 <input type="checkbox"/> 性能测试 <input type="checkbox"/> 接口测试 <input type="checkbox"/> 容量测试 <input type="checkbox"/> 可恢复性测试 <input type="checkbox"/> 余量测试 <input type="checkbox"/> 敏感性测试 <input type="checkbox"/> 可靠性测试 <input type="checkbox"/> 安全性测试 <input type="checkbox"/> 数据处理测试 <input type="checkbox"/> 边界测试 <input type="checkbox"/> 强度测试 <input type="checkbox"/> 互操作性测试 <input type="checkbox"/> 安装性测试 <input type="checkbox"/> 兼容性测试 <input type="checkbox"/> 人际交互界面测试		
需求追溯	软件研制任务书 4.1.6 软上电功能 软件需求规格说明 3.2.2 软上电的功能需求		
初始化	交流起动机控制器各设备正常连接。		
输入和步骤	步骤 1：系统上电，软件完成初始化； 步骤 2：通过外部电源调节母线电压值为 191.5V，确认 1s 后，根据仿真环境观察 K5 软上电继电器是否处于断开状态（即为 1），晶闸管是否属于开启状态（即为 0），软上电完成标志是否为 1。		

4 实验结果

为更加全面评价本文基于语义匹配模型的软件测试用例共享和复用方法的有效性，将本文方法与朴素贝叶斯分类模型进行对比，从精准度、召回率与F1值三个方面对问题模板匹配精度进行评价。F1值代表精准度与召回率的调和平均数。不同问题模板数量的精准度、召回率与F1值如表2所示。

表2 精准度、召回率与F1值

问题模板	本文方法			朴素贝叶斯分类模型		
	精准度	召回率	F1	精准度	召回率	F1
1	91.51%	84.15%	87.64%	86.54%	78.46%	82.30%
2	92.10%	83.64%	87.67%	88.46%	76.56%	82.08%
3	88.16%	83.13%	85.57%	86.13%	77.48%	81.58%
4	95.06%	83.26%	88.77%	94.68%	77.11%	85.00%
5	91.65%	71.64%	80.42%	84.98%	63.46%	72.66%
6	97.98%	95.46%	96.70%	93.46%	86.95%	90.09%
7	94.98%	94.35%	94.66%	93.58%	87.13%	90.24%
8	93.12%	83.46%	88.03%	91.56%	74.61%	82.22%
9	98.14%	83.64%	90.31%	96.45%	72.46%	82.75%
10	93.46%	84.13%	88.55%	90.15%	71.56%	79.79%
11	96.95%	84.56%	90.33%	90.14%	76.46%	82.74%
12	91.56%	71.35%	80.20%	87.56%	61.64%	72.35%
13	89.46%	83.46%	86.36%	89.64%	76.45%	82.52%
14	95.24%	94.64%	94.94%	93.45%	84.13%	88.55%

根据表2可见，无论多少个问题模板，本文方法的精准度均高于朴素贝叶斯分类模型，朴素贝叶斯分类模型的平均精准度为90.48%，本文方法的平均精准度为93.53%，精准度平均提高了3.04%；本文方法的召回率都高于朴素贝叶斯分类模型，朴素贝叶斯分类模型的平均召回率为76.03%，本文方法的平均召回率为84.35%，召回率平均提高了8.32%。本文方法的F1值都高于朴素贝叶斯分类模型，朴素贝叶斯分类模型的平均F1值为82.49%，本文方法的平均F1值为88.58%，F1值平均提高了6.09%。由此可见，本文方法能够更加准确的将用户所需测试用例与数据库中测试用例相匹配，高质量地完成软件测试用例共享和复用。

实验检验本文采用的知识图谱的检索效率，将其与传统测试用例库方法进行对比，两者所用检索时间如表3所示。

根据表3可见，知识图谱所消耗的时间比例均

表3 检索时间

问题模板	时间比例	
	知识图谱	传统用例库
1	1.64	3
2	2.21	3.26
3	1.91	4.56
4	1.92	3.21
5	2.26	3.86
6	2.21	2.92
7	1.95	2.56
8	2.29	3.81
9	2.13	3.26
10	2.43	4.15
11	2.35	3.64
12	1.86	3.76
13	2.04	3.15
14	2.15	3.62

低于传统用例库，因此可以说明本文方法的检索效率高于传统测试用例库。

实验通过对本文方法进行模拟，图3为用户输入的属性信息，图4为本文方法挖掘的相似度最大的软件测试用例。

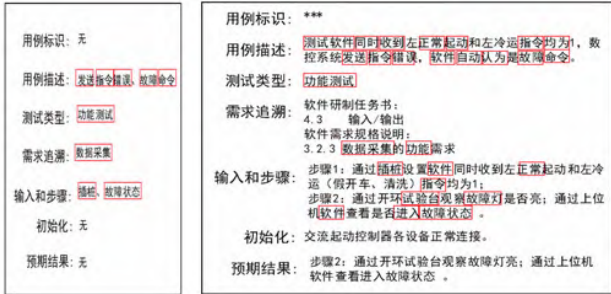


图3 用户输入的属性信息 图4 软件测试用例

根据图3可见，用户输入属性中关键字包括“发送”“指令”，本文方法根据用户输入的关键字搜索到与其相似度最高的软件测试用例，由图4可见该软件测试用例几乎包括了用户输入的全部关键字。说明本文方法能够基于用户提供的属性信息检索到相似度最高的软件测试用例用于分享与复用，提高软件测试效率。

5 结语

对于传统测试用例方法与朴素贝叶斯分类模型的不足，本文通过知识图谱与语义关联性实现软件测试用例的共享和复用。从知识图谱中找到相似的测试用例，进行匹配和推荐，同时也可以根据测试场景、测试类型等因素进行测试用例的筛选和排序，选出最适合的软件测试用例。经过实验验证本文方法的精准度平均提高了3.04%、召回率平均提高了8.32%、F1值平均提高了6.09%，并且缩短了软件测试用例检索的时间。基于语义匹配模型的软件测试用例共享和复用方法，为软件测试用例的复用提供了新思路。

参考文献

[1] 杨波,何宇泽,许福,等.利用改进遗传算法的软件故障定位辅助测试用例生成方法[J].北京航空航天大学学报,2023,49(09):2279-2288.

[2] 夏传林,郑巍,谭莉娟,等.基于知识图谱的测试用例复用方法[J].计算机工程与设计,2022,43(05):1273-1279.

[3] 苗红,郭鑫,李欣.基于产业特征语义匹配模型的产业融合预测研究[J].软科学,2021,35(07):16-24.

[4] 徐浩然,王勇军,黄志坚,等.基于前馈神经网络的编译器测试用例生成方法[J].软件学报,2022,33(06):1996-2011.

[5] 田星雨,曾广迅,高云博,等.基于语义匹配与组合的模型重用技术研究[J].系统仿真学报,2021,33(12):2901-2910.