

密级：公开

# “十三五”全军共用信息系统装备 预先研究项目建议书（会议评审）

项目编号：31511110310

项目名称：基于知识图谱的复杂装备软件

智能化测试技术

申报单位：北京计算机技术及应用研究所

南京大学

东南大学

申请 人：柯文俊

编制日期：2019 年 12 月 20 日



# 目 录

一、军事需求与国内外现状 .....	1
(一) 军事需求 .....	1
(二) 国内外现状分析 .....	4
二、研究目标、研究内容与技术指标 .....	14
(一) 研究目标 .....	14
(二) 研究内容 .....	15
(三) 关键技术 .....	25
(四) 技术指标 .....	27
三、拟采取的研究方法及途径 .....	32
(一) 总体方案 .....	33
(二) 技术途径 .....	138
(三) 创新点 .....	155
四、研究进度、成果形式及应用方向 .....	156
(一) 研究进度 .....	156
(二) 项目成果 .....	158
(三) 应用方向 .....	160
五、经费概算 .....	161
(一) 按研究内容分解经费 .....	161
(二) 按年度分解经费 .....	162
六、研究条件及保障措施 .....	163
(一) 本单位研究基础 .....	163
(二) 人才条件 .....	173
(三) 研究条件 .....	175
(四) 外协条件 .....	177
(五) 管理保障 .....	178



## 一、军事需求与国内外现状

### (一) 军事需求

中央军委主席习近平指出：武器装备是军队现代化的重要标志，是国家安全和民族复兴的重要支撑。面对新形势新任务，装备建设要坚持质量至上，把质量问题摆在关系官兵生命、关系战争胜负的高度来认识，贯彻质量就是生命、质量就是胜算的理念，建立质量责任终身追究制度，着力构建先进实用的试验鉴定体系，确保装备实战适用性。

军用软件是现代武器装备的灵魂，军用软件建设是军队实现信息化的关键。随着军队机械化、信息化水平的逐步提高，武器装备高度集成、软硬件相互渗透化。FPGA、SOC/SIP 广泛应用，精准制导控制、数字信号处理、图像处理及目标识别等领域软件应用密集，DSP 与 FPGA 构件的紧耦合系统得到大量应用，单设备多配置项等因素快速增加，软硬件高度集成，导致武器装备软件密集型特征日益凸显，当下软件质量也已经成为影响武器装备产品质量的核心因素。国防装备软件检测经过 20 年的发展走出了一条自主创新、与国际标准接轨的道路。发展新型软件测试技术，通过知识图谱、人工智能等先进技术对多年积累的装备软件评测经验进行有机总结和运用，让武器装备系统更安全，让软件测试更高效，是履行新时代“强军首责”的首要目标。

1、第三方软件测评作为武器装备软件研制过程中大规模信息汇聚的环节，构建软件测试知识图谱可链接多学科知识，实现装备研制和测试经验总结和复用。

武器装备具有多学科专业交叉群集，多领域技术融合集成的特征，以航天导弹武器为例，包含了导弹弹体设计、发射控制、推进控制、制导控制等多学科专业，需要经过大规模的仿真实验和集成实验才能确定每个子系统的战技指标，其制造是非常复杂且严谨的。在武器装备研制过程中，各学科算法模型和信息交互都会在装备软件中得到具体实现，第三方软件

测评作为保障装备软件质量的重要环节,被纳入到型号研制的初样、试样、定型等多阶段,汇集了大规模的武器装备研制过程的全套数据,有效地收集、整理并充分利用这些数据对武器装备的快速迭代研制、质量把控是很有意义的。

传统关系型数据库和信息检索等技术都难以实现这个目标。知识图谱作为一种人工智能前沿的新型的知识建模和表示技术,能有效从海量武器装备软件研制数据中提取高价值信息及其关联,形成链接多学科知识和装备软件研制全过程的知识图谱,实现装备研制和测试经验总结和复用。

**2、软件定义装备明确了软件在武器系统中的核心地位,也导致装备软件规模的急剧增长,亟需借助知识图谱挖掘装备软件的演化特征,高效保质地引导装备软件进行测试检验。**

随着武器装备网络化、自动化、智能化程度的不断提高,软件规模和复杂度呈几何级数增长。以某导弹的定型测试统计为例,基本型、A型、B型的软件规模分别为:24万行、74万行、270万行;相对于基本型,B型软件规模增长了11倍。随着装备更新速率不断加快,各软件评测机构承担的装备软件测试任务数量快速增长。有关数据显示,某集团下属所有装备软件评测机构的评测型号任务数从2015年到2018年分别为96、129、132、146个,三年中软件型号数量增加了超过50%。如何响应用当下武器装备软件大规模快速交付需求,高效保质地完成装备软件测试检验任务成为棘手的问题。

软件是共性技术。通过分析航空、航天、船舶等国防装备领域的软件特点,在控制模型、业务流程等方面存在较多的共性特征和演化规律,因此相应的软件评测经验也可以相互传承和借鉴。通过知识图谱将装备软件测试数据分解为细粒度的测试工件片段,一方面,在软件测试各环节实施的阶段性测试工件推送,可加速软件评测的效率;另一方面,采取可解释

性的缺陷导向式的测试工件推送模式，以牵引软件测试关注到更关键、更脆弱的功能模块，保障软件测试的质量。

3、武器系统朝着“装备体系化、体系智能化、智能实战化”的方向发展，新的试验鉴定条例要求软件测试从保证质量为主向评估作战效能为主转变，知识图谱是支撑武器装备进行系统级测试的有力手段。

未来武器系统朝着“装备体系化、体系智能化、智能实战化”的方向发展，强调通过武器装备综合运用，形成体系作战能力来指导武器装备的研发、配备和部署。与此同时，武器装备软件的功能性、可靠性、易使用性、效率、维护性和可移植性等特有质量特性在武器装备功能实现和效能发挥上作用越来越突出，由过去执行完成专门硬件的简单功能跃升为保证装备可靠运行和人工智能的逻辑中心和“神经中枢”，直接关系到武器装备战斗力能否有效发挥，其对系统级的功能安全性和可靠性也提出了严格的质量检测要求。

软件测试知识图谱是对历史经验进行有机整合形成的，内部蕴含了导弹系统中如导引头、飞控、测发控等各学科软件的演化规律和协作模式，可辅助测试工程师理解武器装备系统全貌，快速构建武器装备多学科的系统级应用场景和对应的测试充分性评价准则。因此，知识图谱是支撑武器装备进行系统级测试的有力手段，可以为未来武器装备系统级测试奠定基础。

4、装备软件的故障率、返工率一直居高不下，未对故障原因进行深入细粒度剖析，借助测试知识图谱进行大数据分析可识别和发现各类质量要素之间的潜在联系，避免惑于假象而不能深入问题本质的现象。

现有的软件研制模式下生产出的武器装备软件的故障率、返工率一直居高不下。抽出 2010 年以来某军工集团公司各院、基地、公司级发生的软件归零（主要发生在系统联试、靶场飞行试验前）进行分析，发现涉及弹上软件、地面软件、FPGA 等多种类型，软件缺陷暴露时段覆盖了型号

研制各阶段，甚至是定型后的部队演习各个阶段。问题的表象是软件质量不高、可靠性较差，实质却是未对故障原因进行深入细粒度剖析。

借助软件测试知识图谱进行质量大数据分析，识别各类质量要素之间的潜在联系，挖掘影响质量问题的核心要素进行质量度量，并指出影响装备质量的重要环节，从而辅助改进装备软件研制过程，提升软件研制的质量和效率。

## （二）国内外现状分析

### 1. 软件测试国内外研究现状

#### 1.1 国外现状

软件测试技术和软件（程序）密不可分，自从软件问世以来，软件测试就伴随至今。早在 20 世纪 50 年代，英国著名的计算机科学家图灵就曾给出了程序测试的原始定义：测试是正确性确认实验方法的一种极端形式。20 世纪 70 年代，人们逐渐认识到测试在软件开发中具有不可替代的作用，软件测试的研究也开始受到工业界和学术界的重视。

首届软件测试的正式学术会议“IEEE International Test Conference”于 1972 年在美国北卡罗来纳大学召开，是软件测试与软件质量研究和开发人员的第一次聚会，这次会议成为软件测试技术发展的一个重要的里程碑。

20 世纪 80 年代以后，软件测试已经逐步演变成为软件行业一个相对独立的分支。许多和测试相关的行业标准也相继出台：美国国家标准学会（ANSI）和美国电气电子工程师协会（IEEE）在 1983 年制定了 ANSI/IEEE E Std 829-1983 “Standard for Software Test Documentation”，1986 年制订了 ANSI/IEEE Std 1012-1986 “Standard for Software Verification and Validation Plans”，1987 年制订了 ANSI/IEEE Std 1008-1987 “Standard for Software Unit Testing”，1990 年制订了 ANSI/IEEE Std 610.12-1990 “Standard Glossary of Software Engineering

g Terminology”，1998年制订了ANSI/IEEE Std 829-1998“Standard for Software Test Documentation”(Revision of IEEE Std 829-1983)。

目前专门的关于软件测试的国际标准草案已经完成，其标号是ISO/IEC 29119 Software Testing，共包含4个部分的内容 1. 概念和术语；2. 测试过程；3. 测试文档；4. 测试技术。29119将取代的现行国际标准有IEEE 829 测试文档、IEEE 1008 单元测试、BS 7925-1 软件测试的词汇和术语、BS 7925-2 软件组件测试标准。

国际上，软件测试（软件质量控制）是一件非常重要的工程工作，测试也作为一个非常独立的职业。在IBM、Microsoft等开发大型系统软件公司，很多重要项目的开发测试人员的比例能够达到1:2甚至1:4。在软件测试技术方面，自动化测试系统正朝着通用化、标准化、网络化和智能化的方向迈进。

## 1.2 国内现状

我国自80年代开始就意识到软件技术在军用装备开发中的重要作用，为保证军用装备软件质量，我国于1988年制定了GJB 437-1988《军用软件开发规范》、GJB 438-1988《军用软件文档编制规范》和GJB 439-1988《军用软件质量保证规范》3份标准以规范和管理军用软件开发过程。但受我国当时软件技术水平限制，3份标准的实用性并不强，只能起到纲领性目标引导作用，而不能作为软件开发过程的有效指导。在90年代中后期，我国参照MIL-STD-498《军用软件开发和文档》制定了GJB 2786-1996《武器系统软件开发》，并修订发布了GJB 438A-1997《武器系统软件开发文档》使其配套GJB 2786-1996（同时废除GJB 438-1988）。由于来源于实用的美国军用标准，GJB 2786-1996配套GJB 438A-1997的可用性较强，但受限于我国当时软件工程水平的相对落后，在引入MIL-STD-498时理解并不到位，再加上我国军用装备研发体制与美国的差异，该套标准的实施依然存在“水土不服”，与实际的军用软件装备及软件研

发并不十分适应,部分企业在应用标准时经常出现囫囵吞枣生搬硬套的情况。

经过十多年的实践,充分收集了工程实践信息之后,我国于2009年修订发布了GJB 2786A-2009并配套修订发布GJB 438B-2009《军用软件开发文档通用要求》,使用GJB 2786A-2009代替了GJB 437-1988和GJB 2786-1996。至此,GJB 2786A-2009成为全面覆盖军用软件生命周期的软件工程标准,也是目前我国航空工业领域普遍采用的软件开发标准。

我国软件测试技术研究随着软件工程的研究而逐步发展起来。经过多年的发展,国内的软件测试组织已经越来越成熟。在国际软件测试标准制定的基础上,相继制定了许多适合中国特色的软件测试的标准,如GJB/Z 141-2004《军用软件测试指南》、GJB 5234-2004《军用软件验证与确认》、GJB7706-2012《军用嵌入式操作系统测评要求》、GJB 7700-2012《军用数据库管理系统安全性测评要求》、GJB 9433-2018《军用可编程逻辑器软件测试要求》等军用软件测试标准及GJB 6921-2009《军用软件定型测评大纲编制要求》、GJB 6922-2009《军用软件定型测评报告编制要求》等测试文档编写规范,以规范军用软件的测试过程。同时制定了GJB 2725A-2001《测试实验室和校准实验室通用要求》、[2005]装电字第324《军用软件测评实验室测评过程和技术能力要求》等标准与能力要求,指导军用软件第三方评测机构的建立与认证。2004年,总参、总装所属的多家科研单位首批通过全军军用软件测评认证,率先成立了全军第一批软件测评实验室。之后,各大单位也分别成立了相应组织,作为软件测试和评价的专职机构,对确保我军软件质量、提高软件可靠性起到了积极的促进作用。及军用测试实验室建设要求。这些标准的制定为国内的军用软件测试研究提供了有利的平台。

## 2. 软件的智能化测试国内外研究现状

软件的智能化测试即人工智能赋能的自动化测试，其应对的是越来越庞大的研发团队，愈加复杂的研发场景和复杂多变的应用系统带来的软件系统高质量快速交付的挑战。软件测试经历了手动测试、辅助测试、部分自动化、条件自动化和高度自动化的 5 个阶段，正在走向完全自动化，即智能化测试的阶段。当前，软件的智能化测试得到了国内外的广泛关注，取得了大量的研究成果和应用实践。软件的智能化测试重点关注的问题包括智能技术与测试场景的结合、测试用例的自动生成、用例回归、基于人工智能的缺陷定位技术和测试复用等。

## 2.1 国外研究现状

软件的智能化测试对于降低人力成本、提升测试复用和保障软件的高质量快速交付具有十分重要的意义。西方国家在将人工智能应用于软件测试领域已经开展了多年的科学的研究和应用实践，已经研制了一批成熟的应用工具和系统，极大的支撑了软件的智能化测试。

美国的 UFT（统一功能测试）是领先的跨平台测试自动化工具。UFT 为软件应用程序和环境提供回归和功能测试自动化。该工具可以用于自动化桌面、移动、web、net、SAP、ActiveX、Flex、Delphi、Java、Oracle、Visual Basic、PowerBuilder、PeopleSoft、Siebel、Stingray 等许多应用程序。UFT 的特性包括关键字驱动的框架、业务流程测试、健壮的检查点、XML 支持和测试结果。

GUI 和回归测试系统 Squish 以其强大的 IDE 和对主要 GUI 技术的全面支持而闻名。Squish 是 100% 跨平台的工具，可以比较好地支持所有移动、桌面、web 和嵌入式平台。此外，美国的多功能 GUI 测试自动化工具 Ranorex 主要用于移动、桌面和 web 应用程序的端到端测试。通过智能对象识别功能，可以自动检测用户界面的任何变化。该工具的其他突出的特性包括早期 bug 发现、可重用代码模块、易于测试记录、与其他工具的无缝集成以及易于使用的编辑器。

功能测试工具 SoapUI 是广泛使用的用于 SOAP 和 REST api 的开源测试自动化工具，它是 SmartBear 为 API 测试显式设计的功能测试工具。该工具以异步测试、可重用脚本和强大的数据驱动测试而闻名。它附带一个开源的高级版本。高级版本具有一些高级功能，如 SQL 查询生成器、断言向导和表单编辑器。

## 2.2 国内研究现状

中国在软件的智能化测试研究起步较晚，主要由大型的互联网公司、部分研究所和高校在开展相关研制工作，也取得了一定的研究成果和成功的实践。

北京云测 Testin 研制的自动化测试可以在短时间内执行大量的重复性测试任务和多终端测试任务，提供  $7 \times 24$  小时的服务，提高测试效率和产能，有效消除人工执行时的厌烦情绪，确保 App 在功能回归、兼容、性能等各方面的可靠性。其发布的全新 AI 测试产品 iTestin 融合了目前测试领域顶尖的自然语言处理、文本识别、图标识别技术，全面提升测试产品的易用性和自动化效率，在提升脚本编写效率一倍的同时，可以将脚本的维护成本降低了一倍。

腾讯公司在基于人工智能的自动化测试方向也做了很多的尝试，其研制的 QTA Metis 系统主要应用了由海量脚本 UI 逻辑数据进行的大数据、AI 对象识别技术，包含 OCR 以及图像的分类、以及基于人工智能的目标检测技术，可以实时地检测每一个对象，并对每个对象进行分类，极大的提升了 UI 测试的效能。

凯图科技发布的自动化测试工具 KT-ATest 采用独特的图形化方式构建测试场景（测试用例），可以支持测试报告自动生成，具有易于使用与维护的特点，可大大提高测试效率、减少开发与测试费用。

此外南京大学研究发布的众包测试平台慕测 MoocTest 以及阿里公司发布的 Macaca 自动化测试框架等，也都是针对软件测试的一些智能化尝试。

### 3. 知识图谱国内外研究现状

知识图谱的本质是描述实体和实体之间关系的网络，通过它可以对现实世界的事物及其相互关系进行形式化描述。近年来，知识图谱得到国内外的广泛关注，取得了大量的研究成果。知识图谱研究的核心技术问题包括通用和特定领域知识抽取、知识融合、知识表示、知识推理、知识挖掘、知识存储和查询、知识质量评价等。

#### 3.1 国外研究现状

知识抽取包括实体识别和关系抽取等典型任务。在实体识别研究中，随着神经网络模型的引入，出现了一系列使用 RNN 结构并结合 CRF 层的实体识别工作，此类方法无需传统特征工程，使用词向量和字符向量就可以达到很好的识别效果。随后，很多利用预训练的语言模型来完成实体识别的工作提出。Google 提出了 Word2Vec 模型，该模型仅使用单词上下文去预测该单词，具有很好的效果。在 Word2Vec 模型后，又出现了很多优化的方法。在关系抽取研究中，近年来，国外主流的关系抽取研究集中在基于深度学习的关系抽取。

根据知识图谱的应用范围，可分为通用知识图谱和特定领域知识图谱。经典的 Dbpedia、WikiData、Freebase 等都是通用知识图谱。通用知识图谱需要对开放世界的知识进行抽取，需要处理多源异构海量数据源，构建过程费时耗力，无法避免知识缺失，难以快速进行知识更新。基因本体、ICD 医学术语、亚马逊 Product Graph 等都是用于特定领域的知识图谱，虽然没有开放世界的挑战，但也面临着整理领域术语需要专业知识，以及训练数据匮乏等挑战。

在软件工程领域知识图谱中，早在 2007 年，就有研究者对软件工程中的软件工件本体构建进行研究，希望通过构建软件工件本体降低软件维护成本。近年来，现有的软件工程领域实体识别方法主要面向错误报告等单一类型的数据源，或 API 等单一类型的实体，以及单一的软件项目，构建方法不具有通用性和可扩展性，在识别的准确度上未能达到通用领域实体识别的水平。针对于软件安全领域，研究者设计了一种漏洞挖掘算法，通过挖掘三个数据库来分析和获取软件漏洞的基本特征。也有研究者提出了一种深度学习方法，仅使用 CVE 数据集中的漏洞描述来预测软件漏洞的严重性。基于 CWE 数据库中的结构化数据，构建了常见软件漏洞知识图谱，并对其进行嵌入表示，支持软件漏洞相关的各种推理任务。

知识融合领域，用于解决知识图谱间的异构问题，国外研究人员曾对本体层和实例层的匹配展开研究，尝试对 FMA 和 GALEN 两个大规模本体进行匹配，匹配过程采用了一些通用的文本匹配器和结构匹配器，并证明了这种匹配处理的时空复杂度很高。分治思想被应用于知识融合领域，如 Malasco 模型是一种基于分治思想的大规模 OWL 本体影射系统。

知识图谱嵌入领域，用于将知识图谱中的实体、关系、三元组表示为低维稠密向量形式，国外提出翻译模型、语义匹配模型和多源信息融合模型等一系列模型。翻译模型方向将实体之间的关系看做一种翻译操作，并采用基于距离的得分函数来衡量三元组的真实性。语义匹配模型采用基于语义相似度的得分函数以捕获实体和关系潜在的语义关系，主要可分为矩阵分解模型和神经网络模型。多源信息融合模型考虑如何有效地将知识图谱中存在的不同类型的信息表示为向量的形式。

构建的知识图谱质量必然影响后继智能应用效果，国内外研究者近年均意识到该问题的重要性。在知识图谱的质量维度方面，国外提出了一个层次结构框架，从 118 个数据质量属性中归纳出 20 个质量维度，在这些维度基础上，进一步对质量标准和度量标准进行系统整理，重新定义了数

据质量方面的术语、维度、相关衡量指标。在知识图谱数据质量评价方面，一些国外学者尝试从各种角度其标准化，但所提出的划分类型、维度、指标都不相同，还没有形成一个对于数据质量维度的共识。最后，早期的一些数据质量评估工具，如 WIQA、LINK-QA、Sieve 等，并不适用于知识图谱的质量评估。所以，知识图谱质量评估方面还有很多值得探索的问题。

### 3.2 国内研究现状

在知识图谱构建过程中，知识抽取是一个关键环节。目前，从大量文本中抽取知识主要依赖于命名实体识别（NER）和关系抽取等自然语言处理技术。命名实体识别领域，我国早期的研究方向主要是基于统计学习的方法，如隐马尔科夫模型、最大熵模型、支持向量机模型和条件随机场模型等。然而，统计学习方法在较大程度上依赖语料库。随着国外相关研究的深入，我国学者开始将神经网络应用于命名实体识别。2018 年，蚂蚁金服人工智能部开发了基于汉字笔画特征的 cw2vec 模型，提出了新的中文命名实体识别思路。关系抽取领域，我国早期的研究方法是基于知识工程的方法，通过人工编写抽取规则，从文本中抽取与规则匹配的关系实例。近年来，基于机器学习的方法逐渐成为关系抽取的主流思路。

在通用知识图谱上，国内百度、阿里、华为、搜狗等企业均构建了通用知识图谱，用于支撑其业务中的语义搜索等智能应用。东南大学的 Zhi shi.me，复旦大学的 CN-Dbpedia，清华大学的 XLoRe 也都是利用百科数据源构建的通用知识图谱。百度的汉语语言知识图谱、阿里的商品知识图谱、美团大脑等都是国内企业根据自身业务领域构建的领域知识图谱。在软件工程领域知识图谱上，北京大学构建软件工程知识图谱，并通过知识图谱表示学习改善语义搜索效果。复旦大学构建开源项目的 API 知识图谱，关联 API 元素（例如，库，类，方法，参数）、描述性知识（例如，功能，指令）以及相关的背景知识（例如，有关计算机和编程的概念），为软件开发者提供 API 浏览、多功能搜索、API 知识汇总等功能。

知识融合技术的目的是解决知识异构，消除应用系统间的互操作障碍，同时解决本体层和实例层的异构问题。实例层匹配方面，为降低人为干预造成的影响，上海交通大学的研究人员开发了一套半监督学习框架，以自动寻找实例匹配规则。东南大学的研究者利用大规模知识图谱的结构特点和匹配中的区域性特点，提出了一种无需对大规模知识图谱进行分块的知识融合方法，该方法在匹配计算中根据当前得到的匹配结果，以及预测后继相似度计算中可以跳过的位置，从而达到提高映射效率的目的。该方法可同时处理知识图谱中的本体匹配和实例匹配。为降低分块结果的冗余性，可以将属性进行聚类操作。清华大学研究人员提出的一套在大规模实例集上解决实例匹配任务的算法框架，运用多重索引和候选集合，其中将向量空间模型和倒排索引技术结合，实现对实例数据的划分。大规模知识图谱的实例匹配可视为机器学习的一个二分类问题，南京大学的研究者采用半监督学习的自训练方法来解决实例匹配问题。

知识表示学习研究中，我国在 TransE、LFM、SSE 等经典模型基础上实现了较多改进与创新。近年来，知识表示学习技术也被用于实例匹配中，该方法在知识图谱嵌入结果的基础上，将实体匹配视为一个二分类问题，期望学习的嵌入结果具有最大的实体匹配似然。清华大学的研究者提出了一种基于隐马尔科夫随机场的概率框架，以解决作者指代消解问题。东南大学的研究者针对学术知识图谱中的作者指代消解问题，提出了一种半监督的作者指代消解方法。

在知识图谱质量评估方面，国内尚未有公开的系统性的工作报告。随着知识图谱技术在各行业的深入应用，尤其在作战指挥决策、情报分析、军事装备智能化软件等方面，知识图谱质量将直接影响应用效果。

#### 4. 当前方法的问题和局限性

通过上述的国内外现状分析，目前并未形成一整套完整的基于知识图谱的复杂装备软件智能化测试体系，就当前的国外相关技术现状和发展趋势，依然面临一系列的技术挑战：

**(1) 军用软件第三方评测机构质量保证压力巨大，亟需管理与技术创新**

第三方软件测试作为软件测试的一种重要模式，在我国已发展了二十年，为中国软件产业发展做出了重要贡献。但是，在现阶段，研制单位在委托第三方测试时，往往是在软件开发的系统集成和验收阶段，要求第三方发现系统中存在的功能、性能和安全等问题，协助系统验收，而在软件的需求、设计和编码阶段很少涉及，缺少对软件整个生命周期的质量全面把控。另一方面，由于第三方测试机构是在项目后期介入，需求文档可能相对滞后，且缺少后续开发过程中多次需求变更文档，造成测试方在对软件需求的把握上无法做到全面和到位，这对软件测试的过程有一定影响。此外，第三方测试通常时间较短，在这种情况下，由于缺乏高效的第三方测试管理工具，造成测试团队花费在测试管理的精力相对较多，测试人员对系统的测试往往是凭借经验，经验丰富的测试人员的知识和技能难以共享，这不利于整体软件测试水平的提高。

**(2) 尚未形成落地的软件智能化测试解决方案**

当前，国内外在软件智能化测试上的发力主要在于将人工智能的一些技术应用在测试流程中，从而提升自动化测试的能力。但是，目前所有的这些工作都是关注在某个特定测试环节或者是特定场景下的具体测试，这些尝试规模较小，并没有形成一整套的落地智能化测试解决方案。针对特定领域，其通常会有严格的测试标准化流程，如何面向全流程统一部署智能化方案，赋能专有领域的软件测试，形成一整套落地的解决方案，是软件智能化测试方向亟待解决的难题。

**(3) 在知识图谱前沿技术和示范应用上存在差距**

国外知识图谱研究和应用处于前沿引领地位，在通用知识图谱构建、知识表示和建模、知识图谱嵌入、知识推理等方面，一直引领前沿技术，国内则在技术的完善和改进上有较多成果。在知识图谱应用上，随着谷歌和 Palantir 等企业将知识图谱示范性应用于搜索和情报分析等场景，国内近年来的大量知识图谱应用也产生了巨大应用价值，但在应用场景广度和深度上还需要持续探索。在特定领域知识图谱构建上，国内外都面临着领域知识构建代价高的困难，主要包括领域知识抽取困难、模型训练数据稀缺、跨任务跨应用知识抽取中数据标注和模型训练耗时费力等挑战，严重制约着知识图谱在特定领域的应用效果。因此，我国知识图谱研究领域依然需要借鉴学习国外先进技术，并及时将研究成果应用于实际生产应用。

## 二、研究目标、研究内容与技术指标

### （一）研究目标

针对新型智能化测评技术需求，开展基于知识图谱的复杂装备软件智能化测试研究，以海量软件测试文档、记录及代码等测试过程数据为基础，重点突破面向软件测试相关文档及代码的测试工件自动提取与融合、可扩展的装备软件测试知识图谱自动化构建、基于知识图谱的知识快速检索和智能推送、测试用例自动生成、典型装备领域的测试知识图谱质量评价、典型装备软件质量评价等技术，实现复杂装备软件评测产物符合“数据→信息→知识→智能”链条的转化，形成基于知识图谱的复杂装备软件智能化与一体化测试平台，将知识图谱技术应用于复杂装备软件测试领域，解决目前第三方软件评测领域自动化程度不高、测试成果重用率低、历史经验传递困难等问题，推进复杂装备软件测评技术发展，改进传统测试模式，提高复杂装备软件第三方评测效率，从而进一步保障装备质量。

具体包括：

(1) 在装备软件评测关联流程与重要测试过程数据的分析的基础上，基于重用价值、参考价值、可获取性及对智能化测试的增益定义出不低于 10 个测试工件。

(2) 开展面向文档和代码的测试工件自动提取与融合研究，实现软件测试过程文档、记录及代码中测试工件的自动提取与融合，为测试知识图谱构建提供可靠的支撑数据。

(3) 开展软件测试知识图谱自动化构建技术研究，构建出可扩展的复杂装备软件测试知识图谱，同时能够选用合理的评估指标和方法保障知识图谱质量。

(4) 开展知识快速检索、智能推送及测试用例自动生成技术研究，充分利用构建的装备软件测试知识图谱，实现测试知识快速检索、测试工件智能推送与测试用例的自动生成。

(5) 开展装备软件质量评价技术研究，以装备软件测试知识图谱为基础，实现装备软件从时间、院所、型号、领域、阶段等多视角下的软件质量综合评估。

(6) 研制基于知识图谱的复杂装备软件智能化与一体化测试平台，基于构建的装备软件测试知识图谱，能够提供测试知识快速检索、测试工件智能推送与测试用例的自动生成、装备软件质量综合评估等智能化服务。

经过研究，本项目技术成熟度预期达到 5 级。

## **(二) 研究内容**

本项目研究基于知识图谱的复杂装备软件智能化测试技术，旨在形成一套面向装备软件测试过程的智能化测试方法论和解决方案。

本项目研究内容如下图所示，其详细的研究内容包括如下 6 部分：基于装备软件评测流程的测试工件定义与分析、测试工件自动提取与融合技术、软件测试知识图谱自动化构建技术、知识快速检索与智能推送及测试

用例生成技术、典型装备软件质量评价技术、复杂装备软件智能化和一体化测试平台。

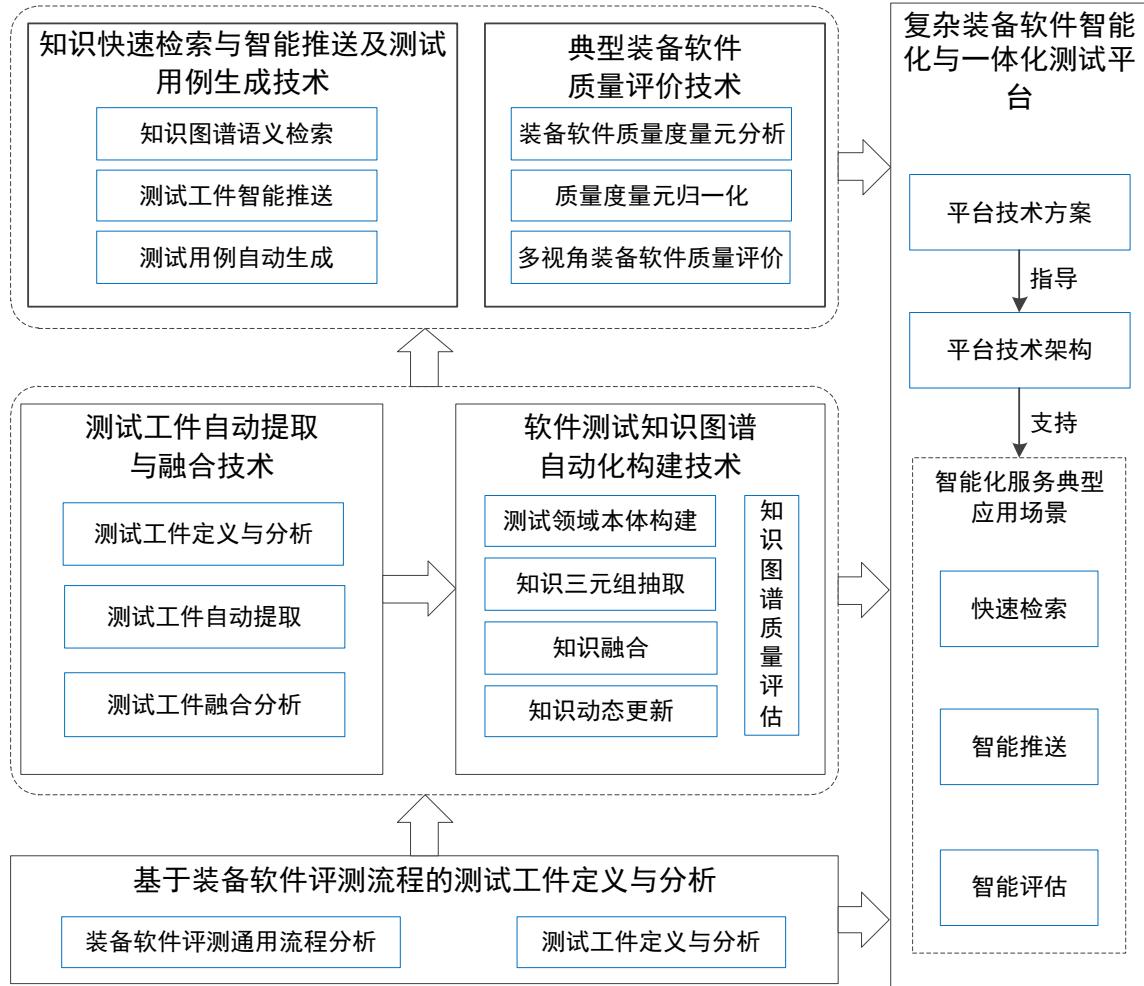


图 2-1 基于知识图谱的复杂装备软件智能化测试技术研究内容图

首先，开展基于装备软件评测流程的测试工件定义与分析技术研究。分析复杂装备软件测试通用流程，梳理软件测试过程各阶段的主要任务及相关数据，进而基于测试过程数据中测试元素的重要程度及对于智能化测试的增益价值进行测试工件的定义，为后续测试工件抽取、软件知识图谱构建奠定基础。

第二步，开展测试工件自动提取与融合技术研究，通过对测试流程的离线历史数据以及作业中的实时流转数据的清洗去除数据中的噪声，提升后续测试工件提取与融合的效率；进而研究基于神经网络的文档类型分

类模型，识别文档所属阶段和类型，并研究基于语义分析的工件结构化信息提取，提取和融合面向文档的测试工件；最后，研究基于切片分析的代码过滤技术，完成程序代码的分解，过滤无关性代码，提取核心相关的代码片段，并基于自然语言处理对软件设计说明进行代码匹配，实现面向代码的测试工件提取与融合。

第三步，开展软件测试知识图谱自动化构建技术研究，支撑军用装备软件智能化测试服务。面向装备软件测试领域，研究构建软件测试领域本体模型的方法，根据结构化的测试工件数据，重点开展面向军用装备软件测试领域知识三元组抽取研究，攻关军用装备软件测试领域知识抽取模型训练数据集匮乏和跨业务知识抽取的数据标注费时耗力等技术难题，同时，研究知识融合技术以解决知识异构问题，进而，发现和处理知识更新，最后，对构建的测试知识图谱质量进行评估，为后继智能化测试服务提供高质量知识保障。

第四步，开展知识快速检索与智能推送及测试用例生成技术研究。在构建的软件测试知识图谱基础上，研究知识图谱语义检索技术，实现测试知识语义检索；研究基于知识图谱的测试工件推送技术，实现基于知识图谱的测试用例可解释性推荐，结合不同测试工件特点实现分阶段测试工件智能推送；研究基于模板的测试用例生成技术，挖掘分析需求文模板以及对应的测试用例模板，实现基于模板的测试用例生成。

第五步，开展典型装备软件质量评价技术研究。研究装备软件质量度量元分析技术，分析定义装备软件质量度量元，结合软件测试知识图谱生成多维度的软件质量评价数据；研究装备软件质量度量元归一化技术，消除不同质量度量元之间的量纲影响，为后续装备软件质量评价奠定基础；研究多视角装备软件质量评价技术，实现从时间、院所、型号、领域、阶段等视角综合评估装备软件质量。

最后，开展复杂装备软件智能化和一体化测试平台建设研究。研究构建一体化的装备软件智能化测试平台，充分利用智能化手段辅助装备软件测试项目管理与测试实施，在现有软件测试流程基础上，形成智能化平台建设方案。首先本确立智能化测试平台建设方案，明确平台的主要功能模块、数据支撑模块以及各个组成部分的相互关系，然后基于平台建设方案与建设要求进行平台技术架构的设计，最后开展智能化服务典型应用场景研究，将智能化服务无缝衔接到底现有测试流程体系中，指导装备软件智能化测试平台的研制，最终形成基于知识图谱的装备软件智能化一体化测试平台。

综上，本项目以知识图谱为核心开展复杂装备软件智能化测试研究。基于测试过程数据，将其中具有重要重用价值与参考价值的测试元素抽象成测试工件，然后通过测试工件提取技术、知识图谱快速构建技术将历史测试知识充分融合利用，形成基于知识图谱的智能化测试平台，将快速检索、智能推送、智能评估等智能化服务无缝集成装备软件通用测试流程，改进传统测试模式，提高测试工作质量和效率，从而充分保障装备质量。

### 1. 基于装备软件评测流程的测试工件定义与分析

测试工件为测试各阶段中产生或构造的、在未来测试项目中具有重用价值或参考价值的测试元素。基于装备软件评测流程的测试工件定义与分析。首先分析当前装备软件测试的通用流程，重点分析各个测试阶段的主要任务及相关数据，以此作为测试工件定义的基础；然后通过对测试过程数据中测试元素的重要程度及对于智能化测试的增益价值进行测试工件的定义。

#### （1）装备软件评测通用流程分析

开展复杂装备软件测试通用流程分析，明确装备软件测试项目建立、测试需求分析、测试策划、测试设计与实现、测试执行、回归测试、测试总结与测试交付等8个评测阶段各自的主要任务及相关成果，将行业标准

与实际测试场景结合起来，形成复杂装备软件测试一般流程，作为装备软件智能化测试的流程主线，也为装备软件测试工件的定义提供支撑。

## （2）测试工件定义与分析

开展装备软件测试工件的定义与分析，基于对装备软件测试通用流程各阶段主要任务和重要过程数据的分析，从重用价值、参考价值、重要性、可获取性及对智能化测试的增益等多方面考虑，对开展装备软件智能化测试的测试工件进行定义。

### 2. 测试工件自动提取与融合技术

研究试工件自动提取与融合技术，旨在从装备软件测试流程文档及代码中提取可复用的测试工件，并对测试工件进行融合，为后续软件测试知识图谱的构建提供数据支撑。首先研究面向装备软件测试流程数据的清洗技术，提炼高可用测试工件相关数据；进而研究面向装备软件测试文档及软件代码的测试工件提取与融合技术，实现测试工件的抽取与融合，从而完成数据到信息的转化。

#### （1）面向装备软件测试流程数据的清洗技术

研究面向装备软件测试流程数据的清洗技术，实现从离线状态的历史数据和实时作业中的流转数据中过滤噪声数据，进而提炼高可用的测试工件相关数据，从而提高测试工件提取效率。针对离线状态历史数据，研究数据清洗及版本确认技术，解决版本演化过程中的数据冗余问题；针对在线流转数据，自动识别数据所属测试阶段及数据类型，进一步判断文档及代码是否可抽取，为后续文档及代码测试工件抽取提供数据支撑。

#### （2）面向装备软件测试文档的测试工件提取与融合技术

研究面向装备软件测试文档的测试工件提取与融合技术，旨在从装备软件测试过程文档中提取测试工件并完成融合工作，实现装备软件测试过程文档从数据到信息的转化。重点研究基于神经网络的文档类型识别技术，以确定目标文档所属测试流程和文档类型，为测试工件的提取奠定基础；

此外,研究基于语义分析的工件结构化信息提取技术,实现从已知类型文档中自动提取工件,并实现基于属性的测试工件的融合。

### (3) 面向装备软件代码的测试工件提取与融合技术

研究面向装备软件代码的测试工件提取与融合技术,旨在分析代码功能模块是否正确响应了软件设计说明。研究基于切片分析的代码过滤技术,完成程序代码的分解,过滤无关性代码,提取核心相关的代码片段,并基于自然语言处理对软件设计说明进行功能代码片段定位。

## 3. 软件测试知识图谱自动化构建技术

研究面向军用装备软件测试领域的测试知识图谱自动化构建技术,旨在从各类复杂测试场景提取软件测试工件数据基础上,自动化识别软件测试实体,发现各类软件测试实体之间的内在联系,构建测试知识图谱,实现软件测试知识图谱的自动更新,并建立软件测试知识图谱评估指标和方法,保障测试知识图谱质量,支撑后继智能化软件测试服务。首先,研究面向军用装备软件测试领域的本体建模技术,形成软件测试知识图谱的抽象形式化规范描述,包括概念、概念层次、关系或属性、公理等。其次,研究从军用装备软件测试结构化和半结构化数据中自动抽取三元组形式的知识,以及从非结构化测试相关文本和代码中进行实体识别和关系抽取。然后,研究面向异构软件测试知识的融合技术,解决软件测试知识在本体层和实例层的异构问题。进一步,研究最小代价软件测试知识演化方法,解决软件测试知识图谱演化中的知识更新代价。最后,研究知识图谱的质量评估方法,对构建得到的软件测试知识图谱进行质量评估。

### (1) 基于本体构建方法论的软件测试领域本体构建技术

研究软件测试知识图谱中的本体构建方法,对军用装备软件测试领域中的概念、属性、概念层次等抽象知识进行统一的本体建模,构建关于测试需求、测试计划、测试术语和词典等软件测试知识和关于飞控、指控、测发控、雷达、导引头等典型装备软件测试知识的本体,通过本体描述软

件测试过程的抽象知识，最后，利用军用装备软件测试工件数据对构建的本体进行验证和修正。

#### （2）基于多策略的军用装备软件测试知识三元组抽取技术

研究面对复杂的军用装备软件测试工件数据的知识抽取技术，抽取测试知识三元组进而构建知识图谱。面对多种形态的测试数据，单一抽取方法无法胜任军用装备软件测试领域知识抽取任务，所以采用多策略的抽取技术分别解决从不同形式的数据中进行知识抽取的问题。拟研究从结构化和半结构化的软件测试数据中自动抽取知识，实现从结构化软件测试数据和半结构化软件测试数据到软件测试知识三元组的自动转换。对于非结构化测试文本，拟研究从文本和代码等非结构化数据中进行实体识别，研究基于预训练语言模型的实体识别抽取方法。为建立实体间的关联，拟研究结合远程监督和深度学习的关系抽取，并引入注意力机制和强化学习降低远程监督中的噪音。针对装备软件测试领域知识图谱构建过程的模型缺少训练数据和跨任务或跨应用场景知识抽取中数据标注和模型训练工作量大等痛点问题，在上述知识抽取工作基础上，拟重点研究基于远程监督和迁移学习技术的知识抽取方法，分别解决训练数据集构建和跨领域数据标注的困难。

#### （3）基于语义子图和约简锚点的测试知识融合技术

研究本体匹配技术，设计面向军用装备软件测试领域的文本匹配器、语义匹配器和实例匹配器等，在此基础之上采用基于语义子图的虚拟文档和相似度传播等技术解决软件测试知识图谱本体层的异构问题。研究基于约简锚点的大规模实例匹配、基于机器学习的大规模实例匹配、实例匹配的分布式并行处理等技术，解决软件测试知识融合中的时间复杂度挑战和空间复杂度挑战。

#### （4）基于最小代价的知识动态更新技术

研究军用装备软件测试知识的更新技术,以最小代价保证测试数据演化过程中的知识及时更新。研究知识演化操作和知识演化策略,实现知识变化捕捉、变化表示、语义变化、变化实施、变化传播、变化传播等演化关键技术,通过知识演化操作定位知识图谱演化位置,根据知识演化影响力分析,研究最小代价知识更新方法,实现软件测试知识图谱演化中的动态知识更新。

#### (5) 基于多阶段多维度的测试知识图谱质量评估技术

研究软件测试知识图谱质量维度的定义、质量标准及度量方法的定义,定义知识图谱质量评估的通用规范和标准,发现并评估知识图谱中由于数据的质量参差不齐导致的信息不一致性、不完整性、错误等一系列质量问题。设计相应的自动化工具评估知识图谱的质量,为提高知识图谱质量提供指导,对于质量不达标的知识图谱,设计对应的优化方法,使软件测试知识图谱更好地满足于实际应用的需求。发现自动化构建的典型装备领域知识图谱的质量问题,提供改进的指导意见,最终提高知识图谱整体质量使之能够满足复杂装备软件智能化与一体化测试平台的使用需求。

### 4. 知识快速检索与智能推送及测试用例生成技术

研究知识快速检索与智能推送及测试用例生成技术,旨在充分利用全链路历史测试数据和软件测试知识图谱,在软件测试知识图谱上实现知识语义检索,实现知识快速检索。进一步,研究基于知识图谱的测试工件智能推送技术,计算测试工件的重要度,结合各阶段测试流程所需知识特点,完成面向测试过程的测试工件智能推送,并重点研究基于知识图谱的测试用例可解释性推荐。最后,研究基于模板的测试用例自动生成技术,挖掘分析测试项、测试用例模板,抽取测试项关键参数,实现基于模板的测试用例生成。

#### (1) 知识图谱语义检索技术

研究知识图谱语义检索技术，在软件测试知识图谱基础上，结合关键词检索和知识表示学习实现测试知识语义检索。具体而言，首先研究基于关键词的知识图谱语义检索方法，基于关键词生成结构化的查询，实现在知识图谱上直接检索知识；此外，研究基于表示学习的知识图谱语义检索技术，通过知识图谱表示学习技术，将知识节点投射到低维连续的向量空间中，从而将相似知识映射到相邻空间内，以实现知识快速检索。

### （2）基于知识图谱的测试工件智能推送技术

研究基于知识图谱的测试工件智能推送技术，采用语义相似度模型计算获取与用户需求相关的测试工件，并计算不同测试工件的重要度，结合软件测试过程的测试工件需求，实现面向测试过程的测试工件智能推送。重点研究基于知识图谱的测试用例可解释性推荐技术，分析软件需求中的关键命名实体信息，采用语义查询挖掘分析出相似的需求节点，并结合命名实体权重、测试用例有无问题单等信息，对测试用例进行相关度排序，从而为测试人员推荐测试用例，以保障测试用例推荐的可解释性。

### （3）基于模板的测试用例自动生成技术

研究基于模板的测试用例自动生成技术，旨在将存储在软件测试知识图谱中的测试知识（测试用例、问题单、全链路数据等）充分利用，分析构建需求模板与测试用例模板，挖掘需求关键信息，结合对应的测试用例模板实现测试用例生成。

## 5. 典型装备软件质量评价技术

研究典型装备软件质量评价技术，旨在充分利用全流程测试链路数据和软件测试知识图谱信息，将被测软件的各阶段输出融合分析，实现多视角的装备软件质量评价。首先，开展软件质量度量元分析技术研究，结合软件知识图谱中测试需求、测试项、测试用例、问题单等测试工件，形成不同维度的装备软件质量度量元数据；进一步，开展装备软件质量度量元归一化研究，消除不同质量度量元之间的量纲影响，从而为后续装备软件

质量评估提供特征；最后，开展多视角装备软件质量评价技术研究，实现从时间、院所、型号、领域、阶段等视角综合评估软件质量。

### （1）装备软件质量度量元分析技术

在装备软件测试知识图谱基础上，本项目结合装备软件测试知识图谱中存储的测试需求、测试项、测试用例、问题单等测试工件信息，分析装备软件质量度量元，抽取多维度的装备软件质量度量元数据。

### （2）装备软件质量度量元归一化技术

研究装备软件质量度量元归一化技术，采用高斯混合模型对度量元进行概率建模，获取不同质量度量元的概率分布，对度量元进行归一化计算，消除不同质量度量元之间的量纲影响，从而为后续准备软件质量评估提供特征。

### （3）多视角装备软件质量评价技术

开展多视角装备软件质量评价技术研究，研究分析不同质量度量元间的相关关系，构建质量度量元相关关系网络，识别出影响软件质量的关键少数质量度量元，根据不同的质量度量元权重分配进行软件质量的整体评估，并能够从时间、院所、型号、领域、阶段等视角综合评估软件质量。

## 6. 复杂装备软件智能化与一体化测试平台

本研究内容主要包括平台建设方案制定、平台技术架构选择及智能化测试服务典型应用场景三个部分。首先开展平台建设方案的设计，明确基于知识图谱的复杂装备软件智能化与一体化平台的主要功能模块、数据支撑模块以及各个组成部分的相互关系；然后基于“需求牵引、简单易用、易于扩展”的建设原则，根据平台建设方案选择适用的平台技术架构；最后基于当前装备软件测试一般过程明确智能化测试服务的典型应用场景，从而指导基于知识图谱的复杂装备软件智能化与一体化平台的建设。

### （1）平台建设方案

智能化测试平台建设方案研究主要任务是确定基于知识图谱的复杂装备软件智能化与一体化平台的功能模块、数据支撑模块以及各个组成部分的相互关系。本方案基于复杂装备软件智能化测试需求，制定了基于知识图谱的复杂装备软件智能化与一体化平台三大模块、一个知识图谱、一个数据中心的基本平台架构。

### （2）平台技术架构

智能化测试平台技术架构研究，主要从开发实现角度明确了平台建设方式。基于知识图谱的复杂装备软件智能化与一体化平台不是一蹴而就的，因而在架构选择时应保证基于知识图谱的复杂装备软件智能化与一体化平台的先进性、可扩展性与易集成性，为此，本项目拟采用面向服务的架构作为基于知识图谱的复杂装备软件智能化与一体化平台的技术架构。

### （3）智能化测试服务典型应用场景

智能化测试服务典型应用场景，基于当前装备软件测试的一般流程与主要测试活动，主要研究装备软件智能化测试服务的实际应用场景。本方案基于不同测试活动的需求差异将智能化测试服务典型应用场景划分快速检索、智能推送与智能评估三种模式，本项目将以三种应用模式为基础，不断挖掘智能化测试需求，为改进传统测试模式，以提高测试工作效率，从而进一步保障装备质量提供支撑。

## （三）关键技术

### 1. 军用装备软件测试领域数据稀缺和跨任务场景的知识抽取技术

在复杂的军用装备软件测试中，存在大量跨任务、跨应用、跨领域的软件测试场景。一方面，领域知识抽取模型训练往往面临目标领域训练数据稀缺（标注稀缺和内容稀缺）和数据标注工作费时耗力的难题；另一方面，跨任务等场景的知识抽取模型需要重新训练，数据标注和训练过程费时耗力，难以适应新型武器装备的快速迭代和反复试验的紧迫研制需求，因此成为军用装备软件测试知识抽取的一个痛点问题。

首先，在领域知识抽取模型训练中引入远程监督技术，通过领域词典，包括军用装备领域词典、军用软件开发词典、军用软件测试词典等领域词典以及军事百科词条等开源数据，自动对软件测试工件的数据进行标注，形成模型训练数据集；与此同时，利用已构建的军用装备软件测试知识图谱，对数据集进行自动标注；在此基础上，利用强化学习降低标注数据噪声，从而解决模型训练数据稀缺的难题。然后，在跨任务、跨应用和跨领域军用装备测试场景中引入迁移学习技术，将已有知识抽取模型的训练数据或模型迁移到新的测试场景中，有效减少数据标注工作量和模型训练时间，提高知识抽取效率，保障新型武器装备的研制需求。上述过程对军用装备软件测试领域的知识抽取训练数据集构建和模型训练速度具有重要价值，因此军用装备软件测试领域数据稀缺和跨任务场景的知识抽取技术是本项目的关键技术。

## 2. 军用装备软件测试领域知识图谱的高效和高质量融合技术

在面向复杂装备软件的大规模软件测试知识图谱构建中，一方面，不同项目组会构建相同领域的软件测试知识图谱，这些知识图谱所涵盖的知识往往存在互相重叠或关联，需要进行知识融合；另一方面，不同测试应用可能需要获取其它软件测试知识图谱所包含的知识，或者联合多个知识图谱以实现更强大的功能，同样也需要进行知识融合。知识融合包括本体层的融合和实例层的融合，前者对本体匹配质量提出了较高的匹配精度和召回率要求，后者除了保证实例匹配结果质量外，还需要面临大规模实例匹配的空间复杂度和时间复杂度的性能挑战。

针对本体匹配，精准描述本体中的概念和关系，构建语义描述文档并计算语义文档相似度，并结合结构匹配技术，得到高质量的本体匹配结果。针对实例匹配，在保证匹配结果质量前提下，采用分治思想和分布式并行处理方式可快速解决大规模知识图谱融合的性能问题。上述过程对软件测

试知识图谱的质量和构建速度均至关重要，因此大规模测试领域知识图谱的高效和高质量融合技术是本项目的关键技术。

### 3. 基于知识图谱的测试用例可解释性推荐技术

测试用例推荐是测试工件推荐的重要内容。相似软件需求对应的测试用例具有一定的相关性，因此，可计算与用户需求相似的需求信息，从知识图谱中获取相似需求对应的测试用例作为推荐。然而，知识图谱中包含了软件所有的测试用例信息，只有少部分的测试用例才是发现软件缺陷的关键测试用例，如何从大量测试用例中找寻与用户需求相关的、更易发现缺陷的测试用例是本项目的难点和重点。

在软件测试知图谱基础上，整理已有的需求、测试项和测试用例数据，分析软件需求中的关键命名实体信息，采用语义查询挖掘分析出相似的需求节点，并结合命名实体权重、测试用例有无问题单等信息，对测试用例进行相关度排序，实现测试用例可解释性推荐。因此，基于知识图谱的测试用例可解释性推荐技术是本项目的关键技术。

## （四）技术指标

### 1. 技术指标

本项目技术指标如下：

（1）支持从非结构化装备软件数据中识别领域标签和特定实体，如战技指标、软件类型等；支持从非结构化装备软件数据中抽取知识三元组；

（2）装备软件测试知识图谱具有可扩展性和自学习性，实体数量不低于 20 万，关系数量不低于 60 万，覆盖飞控、指控、测发控、雷达、导引头等典型装备软件类型；

（3）推送测试工件类型不少于 10 种；

（4）测试过程数据的自动化收集与分析，支持知识图谱的动态更新；

（5）实现基于测试过程数据对装备软件进行质量评价，支持装备软件质量多角度评估与分析；

- (6) 装备软件领域命名实体识别准确率不低于 90%;
- (7) 面向非结构化数据的知识抽取的召回率不低于 85%，准确率不低于 85%;
- (8) 智能推送 TOP3 准确率不低于 90%;
- (9) 典型装备软件质量评价指标不少于 3 种，基于测试数据的装备软件质量评价准确率不低于 80%;
- (10) 支持面向软件文档的测试工件提取与融合;
- (11) 支持面向软件代码的测试工件提取与融合。

## 2、技术指标验证方法与条件

本项目将基于知识图谱的复杂装备软件智能化测试技术应用于装备软件评测中，将软件智能化测试系统性能与技术指标进行逐条对比，实现对技术指标的验证。针对本项目的技术指标，具体验证方法如下。

(1) 技术指标“支持从非结构化装备软件数据中识别领域标签和特定实体，如战技指标、软件类型等；支持从非结构化装备软件数据中抽取知识三元组”的验证方法和条件，如下表所示。

表 2-1 技术指标 1 验证方法和条件

技术指标	研究内容	验证方法	通过条件
支持从非结构化装备软件数据中识别领域标签和特定实体，如战技指标、软件类型等；支持从非结构化装备软件数据中抽取知识三元组	3.2 基于多策略的军用装备软件测试知识三元组抽取技术	<p>(a) 打开软件智能化评测系统，进入知识图谱实体抽取模块，输入含有领域标签和特定实体的非结构化文本信息，查看是否可以正确识别标签和实体；</p> <p>(b) 进入知识三元组抽取模块，输入非结构化测试文本，查看是否可以正确抽取知识三元组。</p>	实体抽取模块可实现从非结构化文本中抽取实体，知识三元组抽取模块可实现从非结构化文本中抽取知识三元组

(2) 技术指标“装备软件测试知识图谱具有可扩展性和自学习性，实体数量不低于 20 万，关系数量不低于 60 万，覆盖飞控、指控、测发控、雷达、导引头等典型装备软件类型”的验证方法和条件，如下表所示。

表 2-2 技术指标 2 验证方法和条件

技术指标	研究内容	验证方法	通过条件
装备软件测试知识图谱具有可扩展性和自学习性，实体数量不低于 20 万，关系数量不低于 60 万，覆盖飞控、指控、测发控、雷达、导引头等典型装备软件类型	3 软件测试知识图谱自动化构建技术	(a) 打开软件智能化评测系统中的知识图谱基础支撑模块，查看并统计实体数量和关系数量信息； (b) 进一步验证不同节点所属典型装备软件类型是否覆盖飞控、指控、测发控、雷达、导引头等典型装备软件类型。	实体数量大于 20 万，关系数量大于 60 万，实体关系所属软件类型包含飞控、指控、侧发控、雷达和导引头等

(3) 技术指标“推送测试工件类型不少于 10 种”的验证方法和条件，如下表所示。

表 2-3 技术指标 3 验证方法和条件

技术指标	研究内容	验证方法	通过条件
推送测试工件类型不少于 10 种	1. 2 测试工件的定义与分析 4. 2 基于知识图谱的测试工件智能推送技术	(a) 打开软件智能化评测系统，统计系统中设计的测试工件类型和数量； (b) 进入测试工件推荐模块，统计可推送的测试工件类型。	推荐的测试工件类型不少于 10 种

(4) 技术指标“测试过程数据的自动化收集与分析，支持知识图谱的动态更新”的验证方法和条件，如下表所示。

表 2-4 技术指标 4 验证方法和条件

技术指标	研究内容	验证方法	通过条件
测试过程数据的自动化收集与分析，支持知识图谱的动态更新	3.4 基于最小代价的知识动态更新	(a) 打开软件智能化评测系统，查看知识图谱模块； (b) 检查知识图谱模块是否可根据测试项目的进行动态更新。	知识图谱可动态更新

(5) 技术指标“实现基于测试过程数据对装备软件进行质量评价，支持装备软件质量多角度评估与分析”的验证方法和条件，如下表所示。

表 2-5 技术指标 5 验证方法和条件

技术指标	研究内容	验证方法	通过条件
实现基于测试过程数据对装备软件进行质量评价，支持装备软件质量多角度评估与分析	5 典型装备软件质量评价技术	(a) 打开软件智能化评测系统，进入软件质量评价模块； (b) 选择不同软件，查看软件多个视角的评价结果。	实现了多个视角的软件质量评价

(6) 技术指标“装备软件领域命名实体识别准确率不低于 90%”的验证方法和条件，如下表所示。

表 2-6 技术指标 6 验证方法和条件

技术指标	研究内容	验证方法	通过条件
装备软件领域命名实体识别准确率不低于	3.1 基于本体构建方法的软件测试领域本	(a) 打开软件智能化评测中的知识图谱支撑模块，输入实体识别的测试数据，执行实体识别，返回 M	实体识别工作正常，且识别准确率不低于

90%	体构建技术 3.2 基于多策略的军用装备 软件测试知识 三元组抽取技术	个识别实体； (b) 根据人工构建的参考结果，计算 M 个识别结果中的正确数为 N，识别准确率为 $N/M*100\%$ ； (c) 判断识别准确率是否不低于 90%。	90%
-----	--	--	-----

(7) 技术指标“面向非结构化数据的知识抽取的召回率不低于 85%，准确率不低于 85%”的验证方法和条件，如下表所示。

表 2-7 技术指标 7 验证方法和条件

技术指标	研究内容	验证方法	通过条件
面向非结构化数据的知识抽取的召回率不低于 85%，准确率不低于 85%	3.2 基于多策略的军用装备 软件测试知识 三元组抽取技术	(a) 打开软件智能化评测中的知识图谱支撑模块，输入非机构化测试数据，执行知识抽取，返回 M 个知识三元组；  (b) 根据人工构建的参考结果，计算 M 个识别结果中的正确数为 N，测试数据中包含 P 个正确的知识三元组，所以可计算得到抽取准确率为 $N/M*100\%$ ，召回率为 $N/P*100\%$ ；  (c) 判断抽取召回率是否不低于 85%，准确率是否不低于 85%。	知识抽取工作正常，且抽取召回率不低于 85%，准确率不低于 85%

(8) 技术指标“智能推送 TOP3 准确率不低于 90%”的验证方法和条件，如下表所示。

表 2-8 技术指标 8 验证方法和条件

技术指标	研究内容	验证方法	通过条件
智能推送 TOP3 准确率不低于	4 知识快速检索智能推送与	(a) 打开软件智能化评测系统，进入知识快速检索与智能推送模	不同测试工件、知识智能

90%	测试用例生成技术、4.1 基于知识图谱的测试用例智能推送技术	<p>块；</p> <p>(b) 选择不同软件、不同测试阶段中的测试工件，记录不同阶段下测试工件推送结果；</p> <p>(c) 判断不同测试工件推送的准确率，计算平均准确率。</p>	推送平均准确率不低于 90%
-----	--------------------------------	--	----------------

(9) 技术指标“典型装备软件质量评价指标不少于 3 种，基于测试数据的装备软件质量评价准确率不低于 80%”的验证方法和条件，如下表所示。

表 2-9 技术指标 9 验证方法和条件

技术指标	研究内容	验证方法	通过条件
典型装备软件质量评价指标不少于 3 种，基于测试数据的装备软件质量评价准确率不低于 80%	5 典型装备软件质量评价技术、5.3 多视角装备软件质量评价技术	<p>(a) 打开软件智能化评测系统，进入软件质量评价模块；</p> <p>(b) 选取不同装备软件，查看软件质量评价指标，并对典型软件集进行质量评价，收集质量评价结果。</p>	装备软件质量评价指标不少于 3 种，质量评测准确率大于 80%

(10) 技术指标“支持面向软件文档的测试工件提取与融合”的验证方法和条件，如下表所示。

表 2-10 技术指标 10 验证方法和条件

技术指标	研究内容	验证方法	通过条件
支持面向软件文档的测试工件提取与融合	2.2 面向装备软件测试文档的测试工件提取与融合技术	<p>(a) 登录基于知识图谱的复杂装备软件智能化与一体化测试平台；</p> <p>(b) 进入面向文档的工件提取与融合模块，查看过程输出结果。</p>	可以查看明确格式的半结构化 json 格式测试工件

(11) 技术指标“支持面向软件代码的测试工件提取与融合”的验证方法和条件，如下表所示。

表 2-11 技术指标 11 验证方法和条件

技术指标	研究内容	验证方法	通过条件
支持面向软件代码的测试工件提取与融合	2.3 面向装备软件代码的测试工件提取与融合技术	(a) 登录基于知识图谱的复杂装备软件智能化与一体化测试平台； (b) 进入面向代码的工件提取与融合模块，查看过程输出结果。	可以查看明确格式的半结构化 json 格式测试工件

### 三、拟采取的研究方法及途径

#### (一) 总体方案

如下图所示，为“基于知识图谱的复杂装备软件智能化测试技术”的总体方案图，主要由“基于装备软件评测流程的测试工件定义与分析”、“测试工件自动提取与融合技术”、“软件测试知识图谱自动化构建技术”、“知识快速检索与智能推送及测试用例生成技术”、“典型装备软件质量评价技术”、“复杂装备软件智能化与一体化测试平台”六部分组成。

具体而言，“基于知识图谱的复杂装备软件智能化测试技术”明确装备软件智能化测试的通用流程，定义各阶段测试工件，为测试工件抽取提供规范指导；“测试工件自动提取与融合技术”实现从软件测试文档及软件代码中抽取测试工件，并进行清洗融合，为后续装备软件测试知识图谱的构建提供数据支撑，实现数据到信息的转化；“软件测试知识图谱自动化构建技术”在抽取的测试工件的基础上构建测试知识图谱，将测试数据有机关联与融合，将信息转化为测试知识；“知识快速检索与智能推送及测试用例生成技术”在软件测试知识图谱的基础上，实现测试工件快速检索与智能推送，同时基于模板生成测试用例，辅助测试人员完成测试，实现测试知识的智能化运用；“典型装备软件质量评价技术”结合软件测试知识图谱中的历史数据，实现多视角的装备软件质量评价；最后，构建“复

杂装备软件智能化与一体化测试平台”，基于装备软件测试知识图谱，在软件测评过程中提供智能化测试服务，有效提高软件评测人员的工作效率和工作质量，从而进一步保障装备质量。

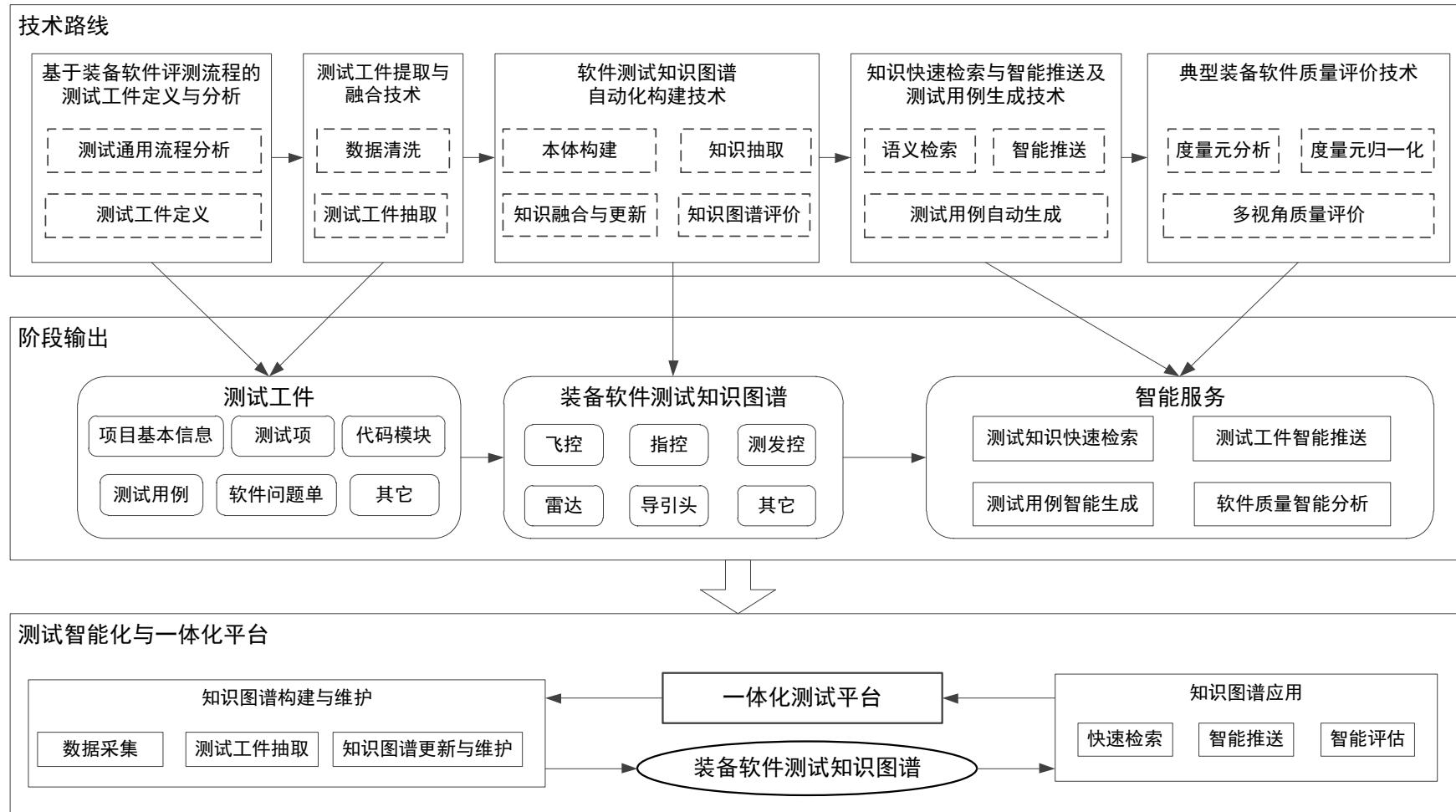


图 3-1 基于知识图谱的装备软件智能化测试技术总体方案图

## 1. 基于装备软件评测流程的测试工件定义与分析

研究复杂装备软件的智能化测试,首先应明确装备软件评测通用流程与评测过程中使用、产生、流转的相关数据。本部分的工作主要包括装备软件评测通用流程分析、测试工件定义与分析两部分构成,如下图所示:

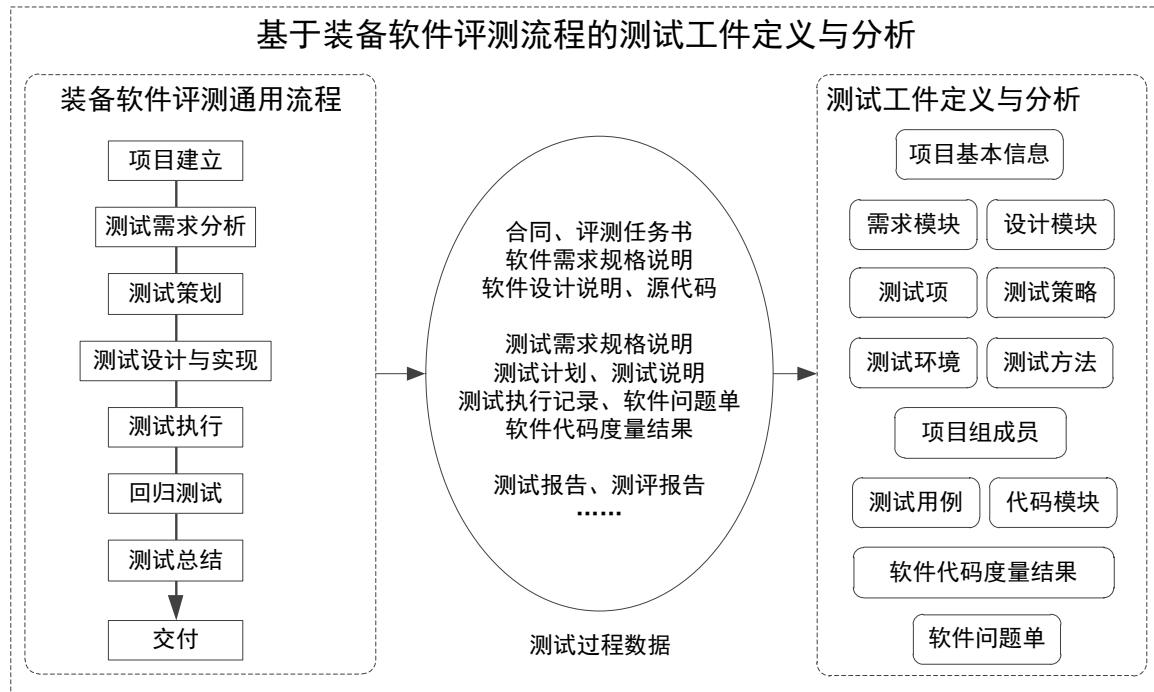


图 3-2 基于装备软件评测流程的测试工件分析与定义

首先,开展复杂装备软件评测通用流程分析,将针对装备软件评测的国家、行业评测要求与第三方测试的实际实施结合起来,明确装备软件在实际第三方测试中的主要阶段、各个阶段的主要任务及相关成果,形成复杂装备软件测试通用流程,作为装备软件智能化测试研究的依据。

然后,通过对装备软件测试各个环节及其中重要的过程产品进行分析,明确各个环节中具有重用价值与参考价值的数据测试数据,进而对测试工件及其内涵进行了定义与说明。测试工件的定义是装备软件智能化测试研究的重要部分,为装备软件智能化测试研究明确了操作对象。

## 1.1 装备软件评测通用流程分析

[2005]装电字第324《军用软件测评实验室测评过程和技术能力要求》中将军用软件测评划分为测试需求分析、测试策划、测试设计与实现、测试执行和测试总结5个阶段，如下图所示。

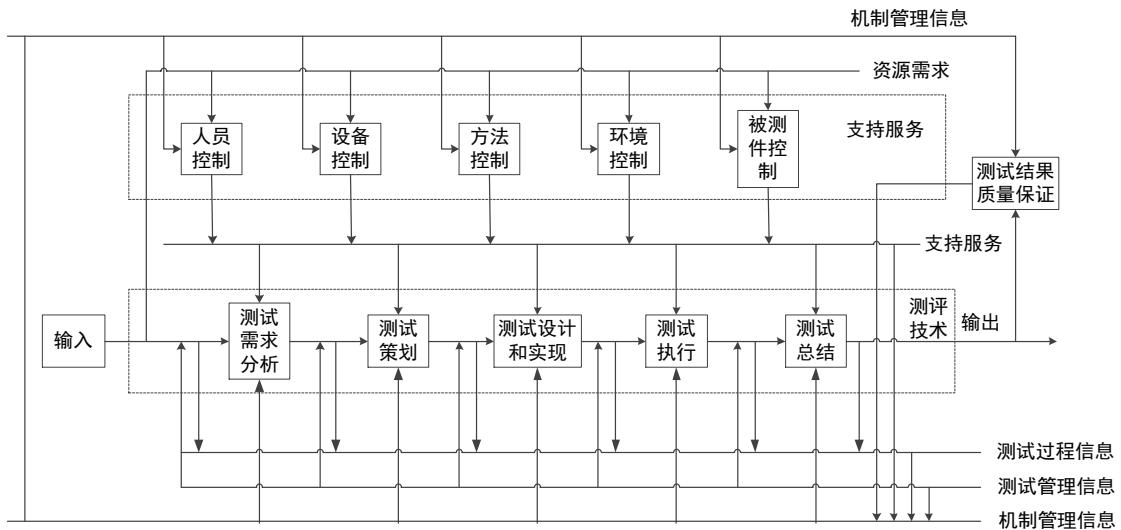


图 3-3 军用软件测评阶段示意图

在实际第三方测试时，测评实验室的承接测试任务后，测试工作以项目形式开展，测试过程一般分为8个阶段，即项目建立、测试需求分析、测试策划、测试设计和实现、测试执行、回归测试、测试总结与交付。实际场景测试流程如下图所示：

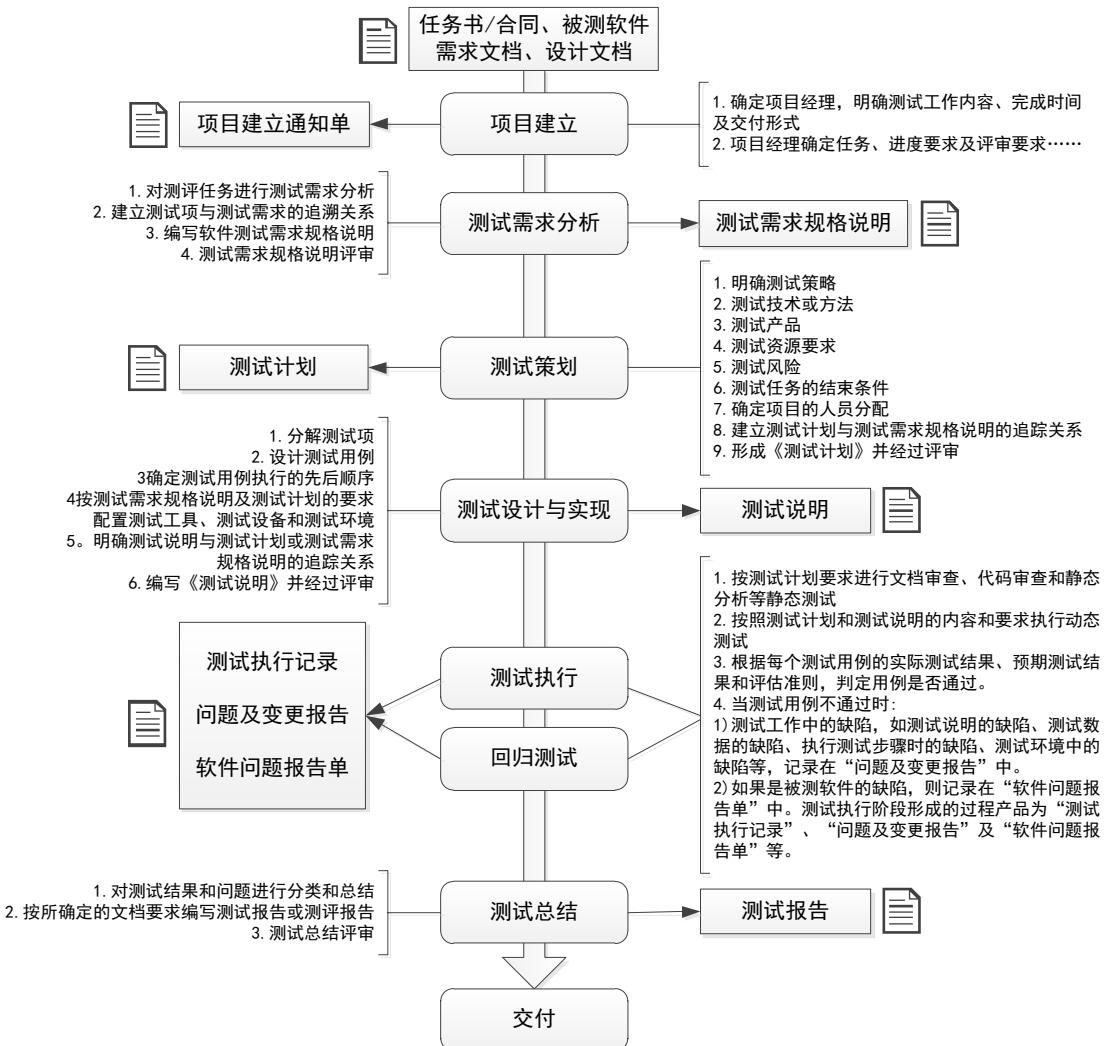


图 3-4 装备软件评测通用流程图

项目建立阶段，测评实验室首先与委托方签订合同或任务书，由实验室负责人基于具体任务确定评测工程部，并安排相关项目负责人了解被测对象的情况，确定测试工作内容、完成时间及交付形式。然后由项目经理确定任务、进度要求及评审要求，建立软件测试项目，填写“项目建立通知单”。

测试需求分析阶段，相关评测人员应首先根据软件测评任务书、合同或其他等效文件，以及被测软件的需求规格说明或设计文档，对测评任务进行测试需求分析，然后基于测试需求分析结果形成测试需求规格说明并建立测试类型中的测试项与软件测评任务书、合同或其他等效文件、以及

被测软件的需求规格说明或设计文档的追踪关系。本阶段形成《测试需求规格说明》并经过评审。

测试策划阶段，相关评测人员应根据软件测评任务书、合同或其他等效文件，以及软件需求规格说明和设计文档等进行测试策划，明确测试策略、测试技术或方法、测试产品、测试资源要求、测试风险及测试任务的结束条件，确定项目的人员分配，形成测试计划并建立测试计划与测试需求规格说明的追踪关系。本阶段形成《测试计划》并经过评审。

测试设计与实现阶段，相关评测人员应根据测试需求规格说明和测试计划进行测试设计和实现，本阶段的任务主要包括分解测试项、设计测试用例、确定测试用例执行的先后顺序、按测试需求规格说明及测试计划的要求配置测试工具、测试设备和测试环境，并编写《测试说明》，在《测试说明》中应明确测试说明与测试计划或测试需求规格说明的追踪关系（本质上是为了建立测试用例——测试项——测试需求的追踪关系），给出清晰、明确的追踪表。本阶段形成的《测试说明》应经过评审。

在测试计划评审和测试说明评审后、测试执行之前，还必须进行测试就绪评审，以确定能否开始执行测试。

测试执行阶段，是测试过程中最核心的阶段，由相关测试人员按照测试计划和测试说明的内容和要求执行测试，包括按测试计划要求进行文档审查、代码审查和静态分析等静态测试；以上审查通过后按照测试计划和测试说明的内容和要求执行动态测试，根据每个测试用例的实际测试结果、预期测试结果和评估准则，判定用例是否通过。当测试用例不通过时，需分析是测试工作的缺陷还是被测软件存在缺陷，如果是测试工作中的缺陷，如测试说明的缺陷、测试数据的缺陷、执行测试步骤时的缺陷、测试环境中的缺陷等，记录在“问题及变更报告”中；如果是被测软件的缺陷，则记录在“软件问题报告单”中。测试执行阶段形成的过程产品为“测试执行记录”、“问题及变更报告”及“软件问题报告单”等。

回归测试阶段，指被测软件更改后，应委托方要求开展的回归测试工作。回归测试中，评测人员应开展软件更改的影响域分析，根据需要修改测试设计。若修改涉及测试方法、测试环境和测试项，则应先修改测试需求规格说明文档、测试计划文档或测评大纲文档，再修改测试说明文档。在此过程中，若测试项、测试用例等变化不超过 20%时，通过文档审签的方式对修改内容进行确认，超过原总量的 20%时，应开展对应的测试需求规格说明或测试说明的评审。审核通过后，则按照测试执行要求开展回归测试。当项目过程中出现多次回归测试时，每次回归测试都应按测试执行阶段的要求实施回归测试，直到满足测试终止条件。

测试总结阶段，评测人员应根据测评任务书、合同、“项目建立通知单”、被测软件文档、测试文档、测试记录、软件问题报告等，对测试工作和被测软件进行分析和评价；根据软件测评任务书、合同（其他等效文件）、被测软件文档、测试需求规格说明、测试计划、测试说明、测试记录和软件问题报告等有关文档，对测试结果和问题进行分类和总结，按所确定的文档要求编写《测试报告》或《测评报告》。在以上各项工作完成后应进行测试总结评审，以确定是否达到测试目的，并给出评审结论。

项目交付，指以上工作完成后，按合同的要求完成交付工作，项目结束。

以上八个阶段，为开展装备软件第三方评测的通用流程，也是本项目开展装备软件智能化测试的依据。

## 1.2 测试工作的定义与分析

装备软件智能化测试旨在通过对历史测试过程数据的深入挖掘，基于历史数据一方面提高测试资产的重用效率，另一方面为在测试过程中提供智能推送服务，辅助测试人员进行测试分析与设计。

装备软件测试过程数据指测评实验室在完成测试任务过程中使用或产生的所有委托方数据、中间数据、产品数据的统称。测试数据是本项目开展智能化软件测试的基础。

测试过程数据主要是指项目测试过程中产生的测试技术相关数据。主要包括如下几类：

- a) 测试前的文档和记录，如：合同、软件测评任务书、被测软件需求规格说明、被测软件设计说明和软件源代码等；
- b) 测试过程中的文档和记录，如：测试需求规格说明、测试计划、测试说明、测试记录、问题报告单和缺陷记录等；
- c) 测评工作完成之后的文档和记录，如：测试报告、测评报告和软件问题报告等。

装备软件智能化测试实现的核心就在于能否从历史测试数据中挖掘出能够在未来测试中重用或参考的重要数据信息。为此本项目引入测试工件的概念。工件原指机械加工过程中的加工对象，在测试场景中，测试工件为测试各阶段中产生或构造的、在未来测试项目中具有重用价值或参考价值的测试元素，如测试项、测试用例、问题单、软件度量结果等。基于此，本项目中定义 12 个关键测试工件，如下表所示。

表 3-1 装备软件评测中关键测试工件

序号	测试工件	测试工件来源（文档或记录）	评测阶段
1.	项目基本信息	合同、任务书等	项目建立
2.	需求模块	软件需求规格说明	测试需求分析
3.	设计模块	软件设计说明	测试需求分析
4.	测试项	测试需求规格说明	测试需求分析
5.	测试策略	测试计划	测试策划
6.	测试环境	测试计划	测试策划
7.	测试方法	测试计划	测试策划
8.	项目组成员	测试计划	测试策划

9.	测试用例	测试说明	测试设计与实现
10.	软件代码度量结果	程序度量结果	测试执行
11.	代码模块	代码模块	测试执行
12.	软件问题单	软件问题报告	测试执行

在上述表格中，《测试需求规格说明》与《测试计划》合并为《定型测试大纲》

### (1) 项目基本信息

项目基本信息为装备软件测试项目的第一也是最基本测试工件，是获取、分析其他测试工件的基础。“项目基本信息”工件至少应包括以下内容：

表 3-2 测试项目信息工件说明

测试工件	属性	属性说明
测试项目信息	项目唯一标识	用于唯一确定项目
	项目类别	普通测试项目、回归测试项目、定型测试项目
	项目密级	非密、秘密、机密
	被测软件名称	被测软件名称
	被测软件版本	被测软件版本
	被测软件类型	飞控、指控、测发控、雷达、导引头等
	研制单位	软件研制单位
	关键等级	A、B、C、D 级
	编程语言	C、C++、C#、Java、Fortran 等

### (2) 需求模块

需求模块是测试需求分析过程中从研制方软件需求规格说明中提取的被测软件需求。“需求模块”工件至少应包括以下内容：

表 3-3 需求模块工件说明

测试工件	属性	属性说明
需求模块	项目唯一标识	项目唯一标识，关联到项目

	需求模块标识	需求模块唯一标识
	需求模块名称	需求名称
	需求模块描述	需求描述
	需求类型	功能需求、性能需求、接口需求等

### (3) 设计模块

设计模块是从研制方软件设计说明中提取的被测软件需求的具体设计。“设计模块”工件至少应包括以下内容：

表 3-4 设计模块工件说明

测试工件	属性	属性说明
设计模块	项目唯一标识	项目唯一标识，关联到项目
	设计模块标识	设计模块唯一标识
	设计模块名称	设计模块名称
	设计模块描述	设计模块描述
	设计及约束	设计及约束
	声明	声明
	输入数据元素	输入数据元素（列表）
	输出数据元素	输出数据元素（列表）
	全局数据元素	全局数据元素（列表）
	局部数据元素	局部数据元素（列表）
	函数调用	函数调用关系（列表）
	追踪关系	与需求的追踪关系

### (4) 测试项

测试项是测试需求分析过程中针对被测软件需求形成的重要产物。测试需求分析过程中应确定测试类型中的各个测试项及优先级，确定每个测试项的测试充分性要求，确定每个测试项应覆盖的范围及范围所要求的覆盖程度，确定每个测试项测试终止的要求。测试项在装备软件测试中具有重要的重用价值与参考价值。“测试项”工件至少应包括以下内容：

表 3-5 测试项工件说明

测试工件	属性	属性说明
测试项	项目唯一标识	项目唯一标识，关联到项目
	测试项标识	测试项唯一标识
	测试项名称	测试项名称
	追踪关系	测试项与研制放需求的追踪关系
	测试项描述	测试项描述
	测试类型	针对此需求有必要开展的测试类型(功能、性能等)
	优先级	测试项测试优先级
	测试要求	测试要求
	测试覆盖\充分性	测试覆盖\充分性说明

### (5) 测试策略

测试策略是测试策划过程中，依据《军用软件测评实验室测评过程和技术能力要求》以及相关国军标的要求，通过分析软件研制任务书和软件需求规格说明等文档，结合软件特点和承研单位提供的测试环境，形成的测评策略，是专家经验的结晶，具有重要的参考价值。“测试策略”工件至少应包括以下内容：

表 3-6 测试策略工件说明

测试工件	属性	属性说明
测试策略	项目标识	项目唯一标识，关联到项目
	策略描述	测试策略描述

### (6) 测试环境

测试环境确定是装备软件测试中的重要环节，主要包括测试工具的选取、软件环境与硬件环境的组成，在装备软件测试中，历史测试项目中测试环境构建方法能够为新项目测试环境构建提供重要支撑，“测试环境”工件至少应包括以下内容：

表 3-7 测试环境工件说明

测试工件	属性	属性说明
测试环境	项目唯一标识	项目唯一标识，关联到项目
	通用测试环境	通用测试环境
	软件测试环境	软件测试环境
	硬件测试环境	硬件测试环境

### (7) 测试技术与方法

测试技术与方法确定是装备软件测试中的重要环节，主要包括测试方法、测试用例生成方法的选取，在装备软件测试中，同类测试项目中采用的测试方法，对于新项目测试方法选择具有重要参考价值，能够大大提高测试方案制定的效率，“测试技术与方法”工件至少应包括以下内容：

表 3-8 测试技术与方法工件说明

测试工件	属性	属性说明
测试技术与方法	项目唯一标识	项目唯一标识，关联到项目
	测试方法	项目测试所选取的测试方法
	测试用例生成方法	测试用例的生成或编写方法

### (8) 项目组成员

选定合适的测试人员组建测试项目组是测试项目管理者的重要工作，对于测试项目的按时交付及交付质量具有重大影响。测试项目中测试项目组成员数据，可以为未来测试项目组成员的选定提供参考，对于管理决策具有重要价值。“项目组成员”工件至少应包括以下内容：

表 3-9 项目组成员工件说明

测试工件	属性	属性说明
项目组成员	项目唯一标识	项目唯一标识，关联到项目
	项目经理	项目经理
	项目组长	项目组长
	项目组成员	项目组成员列表

## (9) 测试用例

测试用例是测试过程中重用率最高的测试组件。历史测试用例，特别是典型的、高质量的测试用例，对于被测软件缺陷的发现具有非常重要的意义。“测试用例”工件至少应包括以下内容：

表 3-10 测试用例工件说明

测试工件	属性	属性说明
测试用例	项目唯一标识	项目唯一标识，关联到项目
	测试用例名称	用例名称
	测试用例标识	用例唯一标识
	测试用例说明	测试用例相关描述
	测试用例初始化	用例初始化方法
	前提与约束	执行测试用例的前提与约束
	终止条件	测试用例执行终止条件
	输入及操作说明	测试用例输入及执行操作说明
	期望测试结果	测试用例正常执行的期望结果
	评估准则	测试用例是否通过评估准则
	设计人员	用例设计人员
	设计日期	用例设计日期
	用例属性	原始、重用

## (10) 软件静态度量结果

软件静态度量结果是装备软件测试过程中静态分析的重要产物，软件规模、圈复杂度、注释率等都是软件质量评价的重要方面。“软件静态度量结果”测试工件至少应包括以下内容：

表 3-11 软件静态度量结果工件说明

测试工件	属性	属性说明
软件静态度量结果	项目唯一标识	项目唯一标识，关联到项目
	源文件个数	软件中源文件个数

	头文件个数	头文件个数
	总行数(包括空行)	代码总行数(包括空行)
	总注释行数	软件总注释行数
	总注释率	总注释率
	文件注释率小于 20% 的比例	文件注释率小于 20% 的比例
	文件注释率	文件注释率
	模块数	软件包含的模块数
	模块圈复杂度平均值	模块圈复杂度平均值
	模块圈复杂度最大值	模块圈复杂度最大值
	模块圈复杂度过大的比例	模块圈复杂度过大的比例
	模块平均行数	模块平均行数
	模块最大行数	模块最大行数
	模块规模(大于 200 行)过大的比例	模块规模(大于 200 行)过大的比例

### (11) 代码模块

程序模块为装备软件测试项目的第一也是最基本测试工件,是获取、分析其他测试工件的基础。“代码模块”工件至少应包括以下内容:

表 3-12 代码模块工件说明

测试工件	属性	属性说明
代码模块	项目唯一标识	项目唯一标识, 关联到项目
	代码模块标识	代码模块唯一标识
	代码位置	X 源文件第 Y 到 Z 行
	与设计追踪关系	X 设计文档第 X 章节

### (12) 软件问题单

软件问题单是装备软件测试过程中的重要产物,软件问题单包括软件问题位置、问题类型、问题级别、问题描述等信息,通过软件问题单的统计分析,能够在软件测试工作中引导测试人员快速把握测试关键环节或关键位置提供重要支撑。“软件问题单”工件至少应包括以下内容:

表 3-13 软件问题单工件说明

测试工件	属性	属性说明
软件问题单	项目唯一标识	项目唯一标识，关联到项目
	问题标识	问题唯一标识
	软件版本	问题对应软件的版本
	报告人	问题报告人
	报告日期	问题报告日期
	问题位置	X 文档的 X 章节或 X 源文件 X 行
	问题个数	问题个数
	问题类别	文档问题、设计问题、代码问题等
	问题级别	关键、重要、一般、建议
	问题描述	问题描述
	设计师意见	修改、不修改、下阶段修改

以上定义了本项目研究中比较重要的 12 个测试工件，12 个测试工件基于项目基本信息工件进行组织管理，基于追溯关系可以实现需求——设计——代码、需求——测试项——测试用例——软件问题单的追踪，同时测试策略、测试环境、测试方法、项目组成员及软件代码度量结果等测试工件，也都是装备软件测试过程中的重要参考。因而，本项目定义的这些测试工件能够较好的为装备软件智能化测试提供支持。

## 2. 测试工件自动提取与融合技术

测试工件自动提取与融合技术总体方案如下图所示，方案体系结构由三个部分组成，分别是：面向装备软件测试流程数据的清洗、面向装备软件测试文档的测试工件提取与融合和面向装备软件代码的测试工件提取与融合。

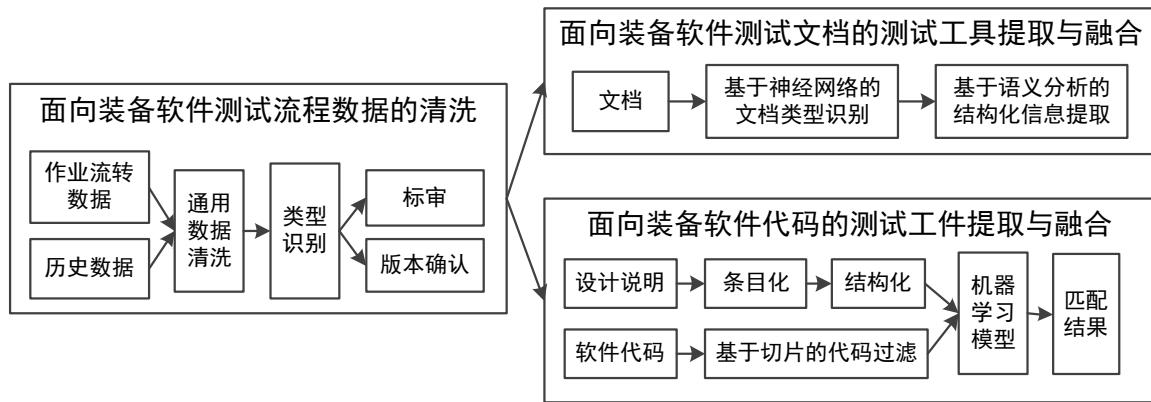


图 3-5 测试工件自动提取与融合技术

首先,针对离线状态的历史数据和实时测试作业中的流转数据进行去噪工作。历史数据都是经过评审后的数据,数据质量较高,但是存在因版本演化带来的数据冗余问题,影响后续的测试工件提取与融合效率,采用最长公共子序列(LCS)算法,依据版本控制系统,对文档做增量计算,识别增量部分,确定版本间关联关系,从而为增量版本数据的测试工件提取提供决策依据。针对作业中的流转数据,主要依据特征关键字识别数据所属类型,进而对其进行标准审查,判断是否满足测试工件提取的要求。

然后,针对面向装备软件测试文档进行相关测试工件的提取与融合,采用神经网络进行文档的类型识别,从而确定目标文档所属测试流程的阶段和文档类型,为测试工件的提取提供依据;此外,使用语义分析手段对工件结构化信息进行提取,实现从已知类型文档中自动提取工件信息,并进一步实现基于属性的测试工件的融合,输出该阶段的 JSON 格式的半结构化测试工件信息。

最后,针对面向装备软件代码的测试工件提取与融合,使用自然语言处理技术对软件设计说明文档进行解析,依次完成条目化抽取和结构化抽取的任务,从而解析出软件设计说明中的关于软件功能模块的实现方法的结构化表示。接下来,采用基于切片分析的代码过滤技术,对程序代码的进行分解,过滤无关代码,提取核心相关的代码片段。构建训练集训练机

器学习模型用于软件设计说明与软件代码的匹配任务，最终输出面向代码的测试工件。

综上，面向软件文档及代码的测试工件自动提取与融合技术首先通过数据清洗解决由版本迭代带来的数据噪声问题，接着使用神经网络判断文档所属的阶段和文档类型，并使用基于规则的和基于语义分析的手段从文档中提取出测试工件相关信息，并经过融合输出半结构化的工件信息，最后采用自然语言处理和切片分析分别对软件设计说明和软件代码进行解析，通过训练机器学习模型完成软件设计说明和软件代码的匹配任务，输出面向软件代码的测试工件。

## 2.1 面向装备软件测试流程数据的清洗技术

文档数据和代码数据都属于文本类型数据的范畴，装备软件测试流程中的相关数据分为离线状态的历史数据和实时作业中的流转数据，这些数据的初始状态一般来说比较“脏”，即经常发生部分字段、属性值缺失的状况，这跟爬虫场景下的数据极其相似：数据的一致性不高、对数据的读写要求高且数据字段会发生变化，针对该数据场景，项目采用非关系型数据库 MongoDB 作为存储方案。针对面向装备软件测试流程数据的清洗，本方案分以下步骤进行。

首要的是针对离线历史数据的清洗工作。首先对文档分词，再使用远程监督方法进行缩略词还原，用开放的百科语料中的专业词和特定领域的词汇表来替换文档中的缩写词汇。之后，进行语法规规范化，错字纠正。缩略词还原，将文档中涉及的缩写补全，并统一选取最长根作为原词。并对语法进行规范化，对测试文档中的语法结构进行一定形式的统一。针对测试文档还需要进行错字纠正，识别出错字并加以修正。使用 NER 方法对时间等部分特定数据进行格式修正，使得不同文件的关键信息含有相同格式，方便后续的测试工件抽取。此外，本项目采用最长公共子序列 (LCS)

算法，依据 Git 版本控制系统，对文档做增量计算，从而识别增量部分，进而依据增量信息确定文档版本的关联关系。

### (1) 采用基于 NLP 方法的内容清洗

基于词典分词算法也称字符串匹配分词算法。该算法按照一定的策略将待匹配的字符串和一个已建立好的“充分大的”词典中的词进行匹配，若找到某个词条，则说明匹配成功，完成对该词的识别。常见的基于词典的分词算法分为以下几种：正向最大匹配法、逆向最大匹配法和双向匹配分词法等。基于统计的机器学习算法，常用的算法有 HMM、CRF、SVM、深度学习等算法。本项目使用机器学习算法和词典分词相结合的方法，一方面能够提高分词准确率，另一方面能够满足领域适应性。

词干和词根化的目的都是将单词的复杂形式及派生相关的形式约减为通用的基本形式。在本项目场景下，对于目标文档，在分词后使用开放语料和专业领域词典，构建缩略词-词根的映射字典，从而实现缩略词还原。此外还对词采用字级别的 n-gram 算法，进行错别字识别和纠正。

### (2) 采用基于 LCS 的增量去重及增量提取

在 Git 版本控制系统中，常使用 LCS 算法来进行 Diff 操作。在本项目中，为更好地从不同的文档中识别出增量信息，进而确定文档间的版本关联关系，减少无效的测试工件提取。采用基于 LCS 的增量文档清洗方法，对不同的文档构建不同的版本层级结构，可以分别对文档增量部分进行迭代的测试工件提取，提高测试工件的提取效率。

在 LCS 算法中，当搜索样本较大时，则容易导致效率低下的问题，为此通常使用动态规划方式解决效率问题，动态规划常常被用来解决一些递归问题，在 LCS 中使用该方法来提升效率。该状态转移方程为：

$$c[i, j] = \begin{cases} 0, i = 0 \text{ or } j = 0 \\ c[i - 1, j - 1] + 1, x_i = y_j \\ \max(c[i - 1, j], c[i, j - 1]), x_i \neq y_j \end{cases}$$

具体的，LCS 可以用于描述两段文字之间的“相似度”，即它们的雷同程度。此外，对一段文字进行修改之后，计算改动前后文字的最长公共子序列，将除此子序列外的部分提取出来，采用这种方法来判断修改的部分。

在识别出文档的增量信息后，对文档进行标记，通过唯一 ID 标识文档，可以对增量内容单独提取并标识。此外，通过 ID 构建文档的分支信息，用于建模文档的演化过程。本项目采用树结构，对文档及其分支进行表示，每个节点表示一个版本，叶节点为最新版本。在测试工件提取阶段，则可分别对不同的节点处理，从而提取完整的工件信息，或依据父节点提取增量部分的测试工件。对于重复的文档，则采用基于树的 DFS 算法，提取增量的工件信息，再获取其父节点的工件提取结果，从而实现对重复文档快速提取。

以上都是针对离线历史数据的清洗工作，该工作在实际生产中主要以跑批的方式实现。

### (3) 基于批处理及实时流处理引擎的数据清洗技术

在装备软件测试过程中需要处理实时作业数据。在对实时作业数据进行清洗时，除了采用前述两种方法分别进行数据纠错、版本清洗之外，还需要对其进行标准审查，以确定是否满足测试工件提取的要求。实时数据其具有产生速度快，处理速度慢的问题，本项目采用实时流处理引擎来处理该问题。对于离线数据的清洗，本项目基于批处理引擎，来进行数据离线处理。

批处理的过程包括将任务分解为较小的任务，分别在集群中的每个计算机上进行计算，根据中间结果重新组合数据，然后计算和组合最终结果。所以批处理系统主要操作大量的、静态的数据，并且等到全部处理完成后才能得到返回的结果。由于批处理系统在处理海量的持久数据方面表现出色，所以它通常被用来处理历史数据，很多 OLAP（在线分析处理）系统

的底层计算框架就是使用的批处理系统。但是由于海量数据的处理需要耗费很多时间，所以批处理系统一般不适合用于对延时要求较高的场景。因而，应用批处理引擎作为离线数据的处理方式，能够胜任本项目历史测试数据的处理工作。

在本项目的测试作业中，需要采用一定的管道技术（Pipeline）实现数据的处理与计算，对每个阶段的实时性有一定的要求，此场景下，若采用批处理技术处理实时数据，则会导致管道瓶颈，降低处理效率。因此，本项目拟采用实时流处理引擎处理实时数据。流处理系统与批处理系统所处理的数据不同之处在于，流处理系统并不对已经存在的数据集进行操作，而是对从外部系统接入的新数据进行处理。流处理系统可以分为两种：一、逐项处理：每次处理一条数据，是真正意义上的流处理；二、微批处理：这种处理方式把一小段时间内的数据当作一个微批次，对这个微批次内的数据进行处理。

项目拟将流处理应用于对实时性要求较高的在线软件测试数据清洗场景。在流处理系统中，除了清洗过程数据外，还进一步地对这些过程数据进行检验，包括对特定文档进行必要字段、属性的校验，以判断其是否满足后续测试工件的抽取条件，从而实现数据过滤，排除噪声数据对抽取任务的影响。

## 2.2 面向装备软件测试文档的测试工件提取与融合技术

项目拟采用 RNN、CNN 方法，分别对文档中的标题、指定域等进行编码，训练多标签分类器用于识别文档所属类型，从而界定目标文档相关的测试工件。根据前序工作中定义的测试工件格式，获取测试工件在对应文档中的既定规则或模板，并利用该规则和模板对文档进行匹配，从中提取到对应的测试工件，并最终输出为 JSON 格式的半结构化数据，包括测试工件定义的不同属性或信息。

### (1) 基于 RNN+CNN 的文档类型识别

RNN 有效地建模了自然语言的场景，对于目标文档能够很好地进行特征提取。但由于过长的文本使用 RNN 时会导致梯度消失，并且无法准确地捕捉特征，因此本项目对文档中标题进行提取。标题通常反映了文档核心类别信息，其他文字均围绕标题展开，因此文档标题能够一定程度代表文档语义。

将词的表示组合成句子的表示，可以采用相加的方法，即将所有词的表示进行加和，或者取平均等方法，但是这些方法没有考虑到词语在句子中前后顺序。使用 LSTM 模型可以更好的捕捉到较长距离的依赖关系，因为 LSTM 通过训练过程可以学到记忆哪些信息和遗忘哪些信息。但是利用 LSTM 对句子进行建模存在一个问题：无法编码从后到前的信息。在更细粒度的分类时，通过 BiLSTM 可以更好的捕捉双向的语义依赖。

本项目采用正则表达式，匹配文档中所有已编号单行文本，以此作为标题项。对于 word 格式的文档，则可以对其格式进行进一步分析，如对其加粗、编号等格式进行匹配，从而提取出标题。本项目拟采用 Bi-LSTM +Attention 的 RNN 结构，在 Bi-LSTM 的模型上加入 Attention 层，并且将最后一个时序的输出向量作为特征向量，然后进行 softmax 分类。Attention 是先计算每个时序的权重，然后将所有时序的向量进行加权和作为特征向量，然后进行 softmax 分类。

## (2) 基于模板的测试工件提取

本部分首先对装备软件测试流程中的文档进行人工的测试工件提取，并编写模板库，测试工件各个元素拆分到不同模板库，以对应不同的文档类型。每个测试工件模板第一个参数为模板库被选中模板的 ID 号，其后的参数为测试工件定义中的各个属性，最后通过使用该模板，对文档内容进行匹配，从而提取到相应的测试工件。

模板在知识图的问答中发挥着重要作用，其中用户话语通过语义解析被翻译成结构化查询。话语模板通常与查询模板配对，指导话语成分到查

询组件的映射。类似地，本项目采用测试工件模板，用以将特定类型文档中的语句同测试工件模板配对，将文档中的各个成分划分到测试工件所需的各个成分中。

模板为一个集合方法，即将工件抽象为 JSON 的数据及其键值对词典，在使用模板对文档进行内容划分时，从文档语句中将若干跨度的词划分到 JSON 的指定键，从而实现基于模板的测试工件提取。同时还可采用规则方法，对由 NLP 方法处理后的文档文本，进行规则匹配，从而提取符合一定语法规则的测试工件。

最终提取到的测试工件，记录为 JSON 半结构化数据，该过程实现了从非结构化文本数据到半结构化信息的转换，提取了文档中可重用的关键信息，为后续的测试知识图谱构建提供可靠的数据。

### (3) 基于属性的测试工件融合

测试工件融合目的在于对相同的工件整合为一个工件，并对不同的工件进行区分。按照测试工件相似度排序，可得一个有序候选工件序列，记为  $t$ ，融合的过程即为对当前工件从有序候选工件中选择可融合工件，对其属性取并集，并将测试工件名进行链接，选择出现频率最高的名称作为测试工件主要名称，其他则为别称。

$$a' = \begin{cases} a \cup arg(t_{max}), t_{max} \geq threshold \\ a, t_{max} < threshold \end{cases}$$

如上式， $threshold$  定义了是否可融合的超参数，而  $t_{max}$  则为序列  $t$  中相似度最大值， $arg(\cdot)$  表示取最大项对象。当相似度大于一定阈值时，则表示该测试工件与候选工件存在较大相似并且能够融合，因此将工件融合，对工件的基本属性取并集。否则，则保留该工件，作为新的独立测试工件，其不可融合。在对可融合工件融合时，存在带融合工件与被融合工件名称不同的问题，本项目则使用名称计数的方式，选择频次高的工件名称作为融合后的测试工件名称。并在后续融合过程中，依次对指定测试工件名进行计数，以更新工件名。相反地，对于名称相同的不同工件，则使用链接

方式进行区分。当最大相似度值小于阈值时，则表示不进行融合，但存在测试工件名相同的问题，为此使用对测试工件添加属性链接至唯一 ID 以区分。本项目充分考虑了融合过程中存在的各种边界情况，来处理可能产生的冲突，并解决该冲突。

进一步地，为提高融合的准确性，项目中拟采用聚类算法，对所有备选项按照相似度聚类，从中准确找出可融合的簇。聚类结果从距离的角度快速地找出了可融合实体间的距离，并增强了关于融合的解释性。

依据前述定义与划分，本部分面向装备软件测试文档的测试工件提取与融合技术研究共输出 11 种测试工件，如下图所示。

```
曰{
    "项目基本信息":⊕Array[9],
    "需求项":⊕Array[5],
    "设计项":⊕Array[12],
    "测试项":⊕Array[9],
    "测试策略":⊕Array[2],
    "测试环境":⊕Array[4],
    "测试技术与方法":⊕Array[3],
    "项目组成员":⊕Array[4],
    "测试用例":⊕Array[13],
    "软件静态度量结果":⊕Array[15],
    "软件问题单":⊕Array[11]
}
```

图 3-6 测试工件示意图

### ①项目基本信息

项目基本信息是装备软件测试项目的最基本测试工件，贯穿整个测试流程，是检索、获取和分析其他测试工件的基础和依据。“项目基本信息”工件从测试合同和任务书这两类文档数据当中进行提取，基本样式如下图所示：

```
曰{
    "项目基本信息":曰[
        曰{
            "项目唯一标识": "00001",
            "项目类别": "定型测试项目",
            "项目密级": "机密",
            "被测软件名称": "Soft",
            "被测软件版本": "1",
            "被测软件类型": "导引头",
            "研制单位": "28",
            "关键等级": "A",
            "编程语言": "Fortran"
        }
    ]
}
```

图 3-7 项目基本信息抽取示意图

### ②需求模块

需求模块是测试需求分析过程中从研制方软件需求规格说明中提取的被测软件需求，从软件需求规格说明文档当中进行提取，其样式如下：

```
曰{
    "需求项":曰[
        曰{
            "项目唯一标识": "00001",
            "需求项标识": "00001",
            "需求项名称": "login",
            "需求项描述": "系统登录",
            "需求类型": "功能需求"
        }
    ]
}
```

图 3-8 项目需求信息抽取示意图

### ③设计模块

设计模块是从研制方软件设计说明中提取的被测软件需求的具体设计，设计模块工件从软件的设计说明当中进行提取，设计模块测试工件样式如图所示：

```
曰{
    "设计项":曰[
        曰{
            "项目唯一标识": "00001",
            "设计项标识": "00001",
            "设计项名称": "login",
            "设计项描述": "系统登录",
            "设计及约束": "*",
            "声明": "*",
            "输入数据元素": "useid",
            "输出数据元素": "null",
            "全局数据元素": "*",
            "局部数据元素": "*",
            "函数调用": "callf",
            "追踪关系": "callLoginReq"
        }
    ]
}
```

图 3-9 设计模块抽取示意图

#### ④ 测试项

测试项是测试需求分析过程中针对被测软件需求形成的重要产物。测试需求分析过程中应确定测试类型中的各个测试项及优先级，确定每个测试项的测试充分性要求，确定每个测试项应覆盖的范围及范围所要求的覆盖程度，确定每个测试项测试终止的要求。测试项是装备软件测试中最重要的测试工件，从测试需求规格说明文档中提取，其样式如图所示：

```
曰{
    "测试项":曰[
        曰{
            "项目唯一标识":"00001",
            "测试项标识":"00001",
            "测试项名称":"testlogin",
            "追踪关系":"*",
            "测试项描述":"登录测试",
            "测试类型":"功能+性能",
            "优先级":"2",
            "测试要求":"登录测试",
            "测试覆盖":"*"
        }
    ]
}
```

图 3-10 测试项抽取示意图

#### ⑤测试策略

测试策略是测试策划过程中，依据《军用软件测评实验室测评过程和技术能力要求》以及相关国军标的要求，通过分析软件研制任务书和软件需求规格说明等文档，结合软件特点和承研单位提供的测试环境，形成的测评策略，是专家经验的结晶，具有重要的参考价值，测试策略从测试计划中提取，其样式如图所示：

```
曰{
    "测试策略":曰[
        曰{
            "项目唯一标识":"00001",
            "策略描述":"*"
        }
    ]
}
```

图 3-11 测试策略抽取示意图

#### ⑥测试环境

测试环境确定是装备软件测试中的重要环节，主要包括测试工具的选取、软件环境与硬件环境的组成，在装备软件测试中，历史测试项目中测

试环境构建方法能够为新项目测试环境构建提供重要支撑，“测试环境”工件从测试计划文档中进行提取，其样式如图所示：

```
曰{  
    "测试环境":曰[  
        曰{  
            "项目唯一标识":"00001",  
            "通用测试环境":"*",  
            "软件测试环境":"*",  
            "硬件测试环境":"*"  
        }  
    ]  
}
```

图 3-12 测试环境抽取示意图

#### ⑦测试技术与方法

测试技术与方法确定是装备软件测试中的重要环节，主要包括测试方法、测试用例生成方法的选取，在装备软件测试中，同类测试项目中采用的测试方法，对于新项目测试方法选择具有重要参考价值，能够大大提高测试方案制定的效率，“测试方法”工件也是从测试计划文档中进行提取，其样式如图所示：

```
曰{  
    "测试技术与方法":曰[  
        曰{  
            "项目唯一标识":"00001",  
            "测试方法":"*",  
            "测试用例生成方法":"*"  
        }  
    ]  
}
```

图 3-13 测试技术与方法抽取示意图

#### ⑧项目组成员

选定合适的测试人员组建测试项目组是测试项目管理者的重要工作，对于测试项目的按时交付及交付质量具有重大影响。测试项目中测试项目

组成员数据，可以为未来测试项目组成员的选定提供参考，对于管理决策具有重要价值。“项目组成员”工件从测试计划文档当中提取，其样式如图所示：

```
曰{  
    "项目组成员":曰[  
        曰{  
            "项目唯一标识":"00001",  
            "项目经理": "*",  
            "项目组长": "*",  
            "项目组成员": "*"  
        }  
    ]  
}
```

图 3-14 项目组成员抽取示意图

### ⑨ 测试用例

测试用例是测试过程中重用率最高的测试组件。历史测试用例，特别是典型的、高质量的测试用例，对于被测软件缺陷的发现具有非常重要的意义。“测试用例”工件从测试说明当中获取，其样式如图所示：

```
曰{
    "测试用例":曰[
        曰{
            "项目唯一标识": "00001",
            "测试用例名称": "testLoginS",
            "测试用例标识": "00001",
            "测试用例说明": "登录步骤",
            "测试用例初始化": "*",
            "前提与约束": "*",
            "终止条件": "*",
            "输入及操作说明": "*",
            "期望测试结果": "*",
            "评估准则": "*",
            "设计人员": "*",
            "设计日期": "*",
            "用例属性": "*"
        }
    ]
}
```

图 3-15 测试用例抽取示意图

#### ⑩软件静态度量结果

软件静态度量结果是装备软件测试过程中静态分析的重要产物,软件规模、圈复杂度、注释率等都是软件质量评价的重要方面。“软件静态度量结果”测试工件从自动化工具跑出来的程序度量结果报告文档中进行提取,其样式如图所示:

```
曰{
    "软件静态度量结果":曰[
        曰{
            "项目唯一标识":"00001",
            "源文件个数": "*",
            "头文件个数": "*",
            "总行数(包括空行)": "*",
            "总注释行数": "*",
            "总注释率": "*",
            "文件注释率小于20%的比例": "*",
            "文件注释率": "*",
            "模块数": "*",
            "模块圈复杂度平均值": "*",
            "模块圈复杂度最大值": "*",
            "模块圈复杂度过大的比例": "*",
            "模块平均行数": "*",
            "模块最大行数": "*",
            "模块规模(大于200行)过大的比例": "*"
        }
    ]
}
```

图 3-16 软件静态度量结果抽取示意图

### ⑪软件问题单

软件问题单是装备软件测试过程中的重要产物,软件问题单包括软件问题位置、问题类型、问题级别、问题描述等信息,通过软件问题单的统计分析,能够在软件测试工作中引导测试人员快速把握测试关键环节或关键位置提供重要支撑,“软件问题单”工件从软件问题报告中提取,其样式如图所示:

```
曰{
    "软件问题单":曰[
        曰{
            "项目唯一标识": "00001",
            "问题标识": "*",
            "软件版本": "*",
            "报告人": "*",
            "报告日期": "*",
            "问题位置": "*",
            "问题个数": "*",
            "问题类别": "*",
            "问题级别": "*",
            "问题描述": "*",
            "设计师意见": "*"
        }
    ]
}
```

图 3-17 软件问题单抽取示意图

以上便是面向装备软件测试文档的测试工件提取与融合的实施步骤和以期达到的研究结果。

### 2.3 面向装备软件代码的测试工件提取与融合技术

首先使用基于语义分析的设计说明条目化描述提取技术，形成条目化的设计说明描述。从设计说明条目化后的文本内容中进一步提取不同实现方法对应的结构化描述，实现设计说明从条目化到结构化的转化。然后基于切片分析的代码过滤技术，完成程序代码的分解，过滤无关性代码，提取核心相关的代码片段；最终通过训练机器学习算法完成基于关键字描述的设计说明和软件代码的匹配。

#### 2.3.1 软件设计说明功能模块设计说明条目化

为了实现软件设计说明与代码的匹配任务，第一步必须实现软件功能模块的设计说明条目化自动提取技术，实现从自然语言描述的非结构化文档中提取设计说明信息，从大量描述语言中抽取真正描述设计的段落、语

句，形成条目化的设计说明信息。自然语言表达的随意性决定自动化处理需求文档的难度大，句型、词性和词义的多种因素影响使得语句变化多样，并且设计种类多、描述不规范，导致很难直接从设计说明文档中提取关于功能模块的实现说明和方法设计。

用户编写软件设计说明文档的方式多种多样，表达习惯也是五花八门、因人而异，但通常关于软件功能模块设计的内在表达逻辑是具有一定规律的，通过借助于数据挖掘的手段可以挖掘出用户表达逻辑的常用模式。本项目首先研究领域分词和词性标注模型，将设计说明文本转化为词性标注序列，进一步研究使用一种简洁、有效的方式对描述中的表达逻辑进行建模，从而抽取出模式，实现软件设计说明的条目化提取。

### (1) 领域分词模型

军事软件系统领域具有自己独特的领域信息，包含特定的领域词典信息，依据大量历史设计说明文档信息，开展基于统计学习的领域分词模型训练，可以实现对中文字符串的准确切分，以支持后续的模式识别和挖掘。这种方法往往不需人工维护规则，也不需复杂的语言学知识，且扩展性较好，是现今分词算法中较常用的做法。

本项目采用多策略的分词模型，结合规则、统计方法、深度学习技术，这样做既可降低统计对语料库的依赖性，充分利用已有的词法信息，同时又能弥补规则方法的不足，而且还可以将深度学习语义表达的有点展示出来。

### (2) 基于统计分析的设计说明模式抽取方法

在对非结构化文档设计说明进行分词标注的基础上，本项目拟对装备软件领域设计说明句式进行挖掘分析，获取常见设计的描述规则，从而构建句型模式。本项目拟采用半自动的设计说明句型模式抽取方法，首先从对分词标注后的领域设计说明中抽取部分领域句式规则，并结合专家意见，人工总结出一些设计表达句型模式；对于每一类句型模式，根据领域需求

描述规律编写正则表达式实现相应的设计表达句型模式；利用设计表达句型模式对新的设计说明进行抽取，根据抽取的准确性来修正设计说明句型模式，同时，若漏掉了一些设计说明句型模式，则将设计说明句型模式和对应的抽取规则加入到抽取规则集合中，若出现抽取错误，则修正相应的句型模式，逐渐获取完善的句型模式。

采用上述半自动化的方式可以实现领域设计说明模式构建，设计说明模式表征设计的撰写逻辑，基于模式构建设计说明抽取模板，可采用模糊匹配方式实现从非结构化文本信息中抽取软件设计信息，获取功能模块设计条目。

在完成了设计说明的条目化之后，接下来基于语义分析完成设计说明条目化到功能实现结构化的转换。采用自然语言处理技术，在设计说明条目化的基础上，结合需求本体模型，抽取每一设计条目不同属性具体对应的功能模块信息，以支持功能模块的结构化描述；并利用关键词提取技术抽取属性关键词，最终实现功能模块设计的结构化表示，为后续设计说明与软件代码的匹配提供支撑基础。

### 2.3.2 软件设计说明功能模块设计说明结构化

需要在前述设计说明条目化的基础上，从某一功能模块的描述中进一步抽取不同功能模块对应的具体实现的设计描述语句，从而实现设计说明从条目化到结构化的转化。本项目拟采用语义分析技术，将设计说明语句进行语义编码，转化为语义向量，并在此基础上训练多分类模型，从而实现设计说明结构化描述。

自然语言处理中常常采用递归神经网络（Recurrent Neural Network, RNN）来编码语义上下文信息，以获取更加深层次的语义信息。为了充分考虑语句前向和反向的语义信息，本项目拟采用双向 LSTM (BiLSTM) 的文本分类模型，结合注意力机制，实现准确、高效的文本分类模型。注意力 (Attention) 机制是自然语言处理领域一个常用的建模长时间记忆机制，

能够很直观的给出每个词对结果的贡献，因此本项目引入 Attention 机制以考虑不同设计说明下具有不同的热点词。双向 LSTM 编码语句的示例如下图：

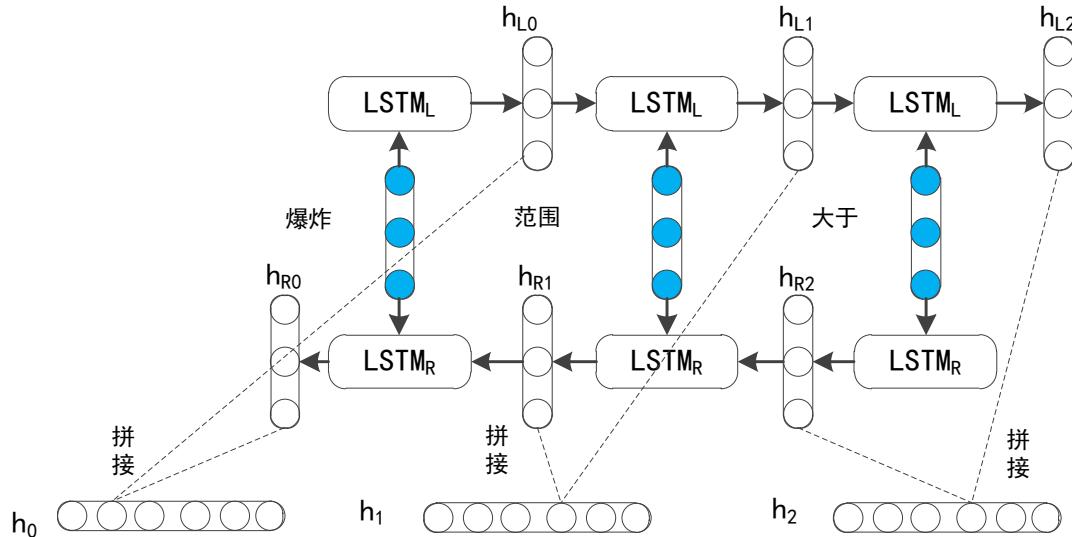


图 3-18 基于 BiLSTM 的语义编码模型

Attention 机制在序列学习任务上具有巨大的提升作用，能够让任务处理系统更专注于找到输入数据中显著的与当前输出相关的有用信息，从而提高输出的质量。

### 2.3.3 基于切片分析的代码过滤

代码切片技术是一种对程序进行识别并提取的技术，根据切片准则，提取源代码中对关注点的变量值有影响的语句，组成新的程序，代码切片指的就是这段程序。代码切片技术可以更好的理解程序、分析程序，现在广泛的应用于软件开发领域及软件应用领域中，在软件测试、维护、程序理解、优化等方面，均具有非常重要的地位。

一个代码切片是由程序中的一些语句和判定表达式组成的集合。关键字驱动的代码切片采用分割思想，它将相关性代码片段和无关性代码分离，获取的代码片段只是特定规则集下程序的抽象描述。代码切片技术通过引入切片，并对其控制依赖和数据依赖进行分析，把需要分析的代码从复杂程序中剥离出来，只提取其中的部分路径和部分模块进行分析，类似于分

治算法，大大降低了代码的复杂度。

针对庞大的被测源代码集合，在进行聚类分析之前需要过滤无关的代码片段，以降低空间和时间开销。本项目采用静态切片的方法，对代码进行预处理，静态切片技术是指在计算程序切片时使用的是静态的数据流和控制流分析方法。该技术对程序的输入不作任何假设，所作的分析完全以程序的静态信息为依据，通过分析程序所有可能的执行轨迹，采用基于文本的过滤方法，利用词法分析技术从基于规则集的代码关键字库中提取相关性关键字，然后扫描源代码，过滤与关键字无关的部分，输出相关性代码片段。

具体的，利用前向程序切片方法，并采用信息流分析技术，基于关注的程序变量，从源代码中提取与该变量有控制依赖关系和数据依赖关系的代码片段，准确了解变量的特征，根据所获得的信息，就能够分析出变量的获取和传播。由于变量间可能存在的相关性，这些代码片段之间有可能存在包含关系，再引用语义线程的思想对代码片段进行合并，从被测代码中切分出一个没有冗余代码片段的最小划分集合。

在完成了以上 3 个步骤之后，便可以初步构建标注数据集训练机器学习的分类模型，训练基于关键字描述的设计说明和代码匹配算法，并提取和融合面向代码的测试工件。

本部分面向装备软件代码的测试工件提取与融合技术研究共涉及 1 种测试工件，命名为“代码模块”测试工件，代码模块主要由软件代码中挖掘计算而来，其样式如图所示：

```
    "代码模块": [
        {
            "项目唯一标识": "00001",
            "代码模块标识": "*",
            "代码位置": "*",
            "与设计追踪关系": "*"
        }
    ]
}
```

图 3-19 代码模块抽取示意图

以上便是面向代码的测试工件提取与融合，可以为后续的测试知识图谱构建提供在代码层面的数据支撑。

### 3. 软件测试知识图谱自动化构建技术

软件测试知识图谱自动化构建技术的总体方案如下图所示，总体方案的体系结构由五个部分组成，分别是：基于本体构建方法论的测试领域本体构建技术、基于多策略的测试领域知识三元组抽取技术、基于语义子图和约简锚点的测试知识图谱融合技术、基于最小代价的知识动态更新技术、基于多阶段多维度的测试知识图谱质量评估技术。

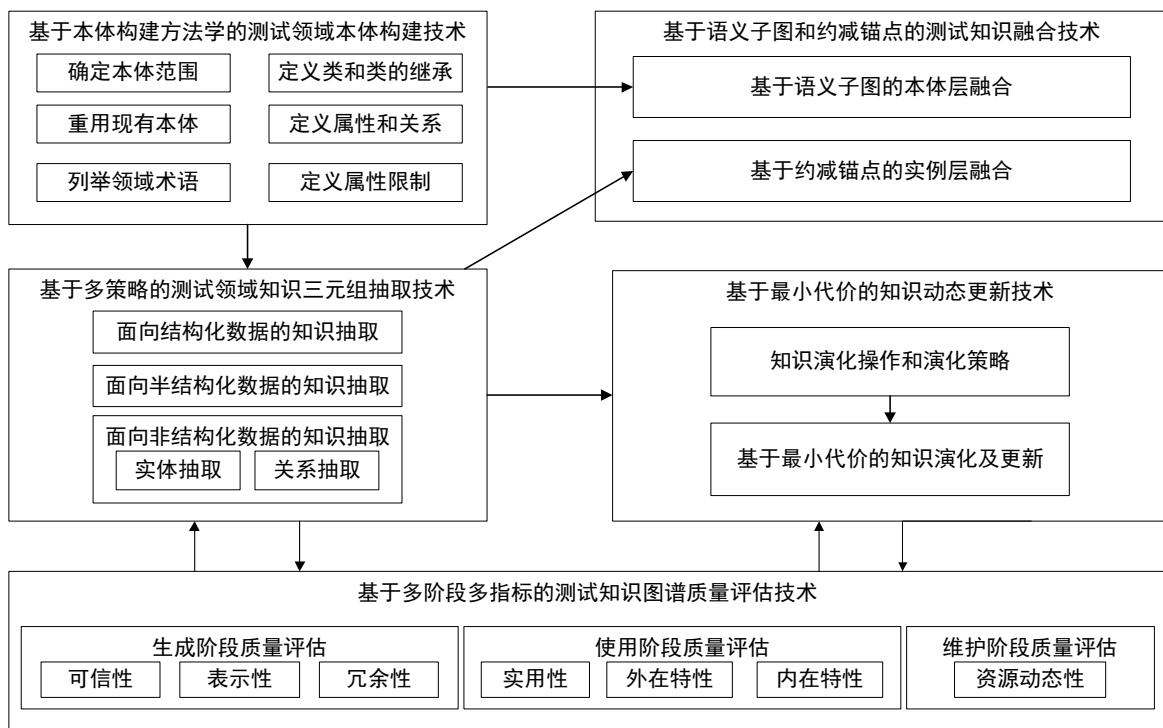


图 3-20 软件测试知识图谱自动化构建技术总体方案

首先，对军用装备软件测试领域中的概念、属性、概念层次等抽象知识进行统一的本体建模，构建关于测试需求、测试计划、测试项、测试用例等软件测试知识的本体，通过本体限定和描述软件测试过程中所产生的知识，并利用测试工件数据对构建的领域本体进行验证和修正。

其次，面向复杂的军用装备的软件测试工件数据，采用多种知识抽取策略，分别从结构化数据源、半结构化数据源和非结构化数据源中进行知识抽取，从而构建软件测试知识图谱，实现从软件测试工件信息到软件测试知识三元组的自动转换。针对军用装备软件测试领域模型训练数据稀缺和跨任务等抽取场景中数据标注工作量大和模型训练耗时等问题，采用远程监督实现数据自动标注，通过迁移学习实现跨领域数据标注。

再次，对不同场景下构建的软件测试知识图谱进行融合，采用虚拟语义描述文档和相似度传播等技术解决软件测试知识图谱本体层的异构问题，采用基于约简锚点的大规模实例匹配、基于机器学习的大规模实例匹

配、实例匹配的分布式并行处理等技术，全面解决软件测试知识融合中的时间复杂度挑战和空间复杂度挑战。

接下来，为了处理军用装备软件测试知识的演化和更新，研究软件测试知识演化操作和知识演化策略，实现知识变化捕捉、变化表示、语义变化、变化实施、变化传播、变化传播等演化关键技术，定位知识演化位置，分析知识演化影响范围，研究最小代价知识更新方法，实现软件测试知识图谱演化中的动态知识更新。

最后，为了保证构建的软件测量知识图谱的质量，分别针对软件测试知识图谱生成阶段、软件测试知识图谱使用阶段、软件测试知识图谱维护阶段的知识图谱特点和场景，选择多个质量评估维度，使用基于分层抽样的增量评估方法对软件测试知识图谱进行质量评估。

总结来说，软件测试知识图谱自动化构建技术通过软件测试领域本体结构指导从软件测试工件数据中自动进行知识抽取，并使用基于语义子图和约简锚点的大规模知识图谱融合技术对构建出的软件测试知识图谱进行融合，同时使用最小代价的更新操作解决软件测试知识图谱的动态更新问题，最后通过多维度多阶段的评估保证软件测量知识图谱质量，获得高质量的知识图谱，形成对基于知识图谱的智能软件测试系统的支持。

### 3.1 基于本体构建方法论的软件测试领域本体构建技术

知识工程领域多年的研究和实践指出知识图谱需要手工构建本体，通过本体限定和描述知识图谱所在领域的知识。知识图谱本体的构建过程通常是迭代的，即先给出初步的本体框架，然后在对本体进行不断修改和精炼的过程中补充细节。本方案软件测试本体创建的过程分为 7 步进行：

步骤 1：确定软件测试本体的领域和范围。首先明确一些基本的问题，包括：软件测试本体的用途是什么？要描述什么信息？回答哪一类的问题？谁将使用和维护这个本体？等等。这些问题可以借助能力咨询的方法来获

得。需要注意的是，随着软件测试的进行，这些问题和回答可能会发生变化，这时需要考虑再回到第 1 步进行迭代开发。

步骤 2：考虑重用现有本体。收集和软件测试本体相关的其它本体是有价值的，例如软件工程本体、装备知识本体、软件需求本体、测试需求本体。一方面如果可以精炼、扩充、或修改现有的本体，那么则可以避免很多不必要的开发工作。另一方面即使现有的本体无法满足当前的应用要求，通常也会从其中得到一些启发和帮助。目前网络上已有一些本体库，从中可以获得很多现有的本体。

步骤 3：列出本体中的重要术语。将所关心的软件测试术语列举出来是非常有用的，例如功能、性能、缺陷、优先级等。这些术语大致表明建模过程所感兴趣的事物、事物所具有的属性和它们间的关系等。这些重要术语能保证最终创建的本体不会偏离复杂装备软件测试领域。

步骤 4：定义类和类的继承。类的继承结构的定义可以采用自顶向下的方法，即从最大的概念开始，然后通过添加子类细化这些概念；也可以采用自底向上的方法，即由最底层、最细的类定义开始，然后找到它们的父类；当然，也可以采用这两种方法的综合进行定义。但无论选择哪种方法，都要从定义类开始，选择那些从具体存在的对象中抽象出来的软件测试术语作为软件测试本体中的类，然后将它们构成分类等级体系。

步骤 5：定义属性和关系。仅有类不能对很多问题给出回答，通常一旦定义了类，还需要定义概念和概念间的内部联系。这里所指的联系可分为两种：一种是概念自身的属性，称为“内在属性”，如概念“测试问题单”的报告日期这种属性可用术语“报告日期”表示，这一类属性通常连接一个概念和一个值，在后面讲述到的本体语言 OWL 中，这种属性被表示为 DataProperty。内在属性具有通用性，也就是说该类对应的所有实例都具有这种属性，并且这种属性通常能向下传递，即如果一个类具有一个内在属性，那么它的所有子类都继承了这种属性。这样也就要求在属性建

模的过程中，一个属性应该为拥有该属性的最大类所拥有。另一类属性称为“外在属性”，也有的文献直接称之为关系，通常用于连接概念间的实例，如概念“测试用例”的一个外在属性“设计人员”连接了概念“人员”，表明对于一对分别来自这两个概念的实例来说，可能会存在“设计人员”这个关系。“外在属性”在随后的 OWL 语言中表示为 ObjectProperty。需要指出的是，步骤 4 和步骤 5 通常同时进行，没有必要严格地分为两步进行，也就是说允许在定义类的同时就定义类所具有的属性和关系。

步骤 6：定义属性的限制。在这一步中，需要进一步定义属性的一些限制，包括属性的基数、属性值的类型，以及属性的定义域和值域。

步骤 7：利用实例验证和优化本体。最后，还需要为类创建实例。这需要确定与个体最接近的类，然后添加个体进去作为该类的一个实例，同时要为实例的属性赋值。在此过程中，验证构建的本体是否能正确和全面描述实例，并对本体进行优化调整，从而形成一个领域本体构建的闭环过程。

在上述 7 个步骤中，术语命名也需要遵循一定准则。在本体中规定合理的命名规则并严格地遵循它们，不仅会使本体易于理解和易于阅读，而且能避免建模中不必要的错误。综上所述，尽管知识图谱本体层建模需要遵循一定的准则和方法学，建模过程有着一些通用的步骤，但是不能忘记本体建模本身就是一个创造性的工作，在遵循一定的工程方法条件下，还需要发挥领域专家的创造灵感。

本方案可构建描述通用软件测试过程知识、软件项目管理知识等能力，一些通用软件测试术语（概念和关系）如下表列举所示，对应的可视化本体图如下图所示。

表 3-14 通用软件测试本体术语

本体概念	关系属性（定义域，关系，值域）
测试项目、测试用例、软件、人员（设计	（项目组，包含，人员）、（设计人员，继

人员、测试人员)、项目组、设计文档、  
测试问题单、需求文档、测试需求、测试  
用例、测试记录、问题报告单、软件、项  
目组、测试报告

承, 人员)、(测试人员, 继承, 人员)、  
(设计人员, 设计, 软件)、(测试项目,  
测试, 软件)、(测试人员, 参与, 测试项  
目)、(测试人员, 设计, 测试用例), (测  
试人员, 执行, 测试用例)

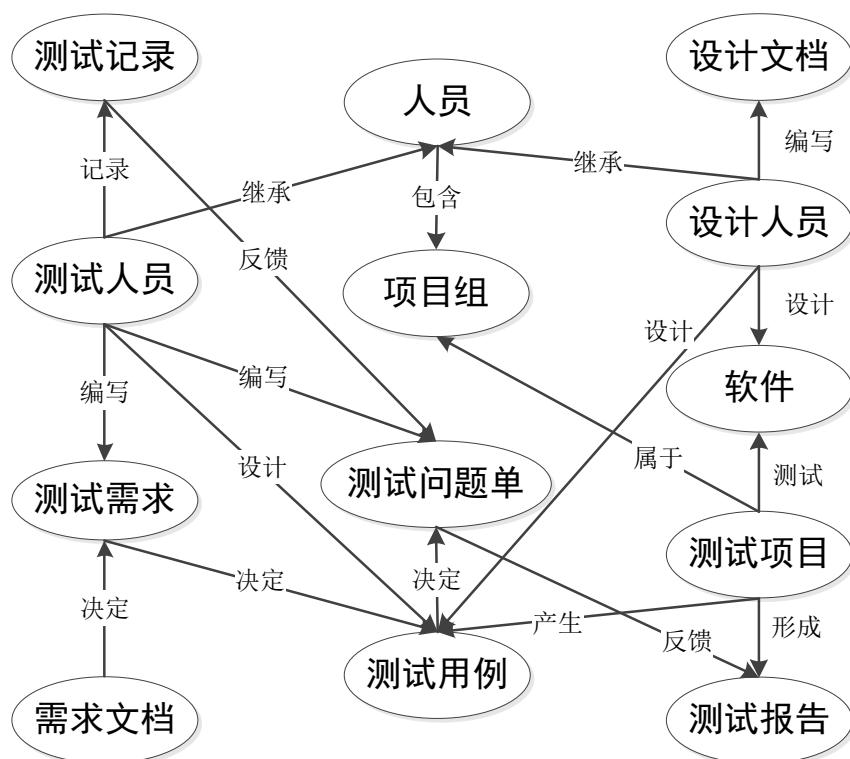


图 3-21 通用软件测试本体图

在上述本体构建方法学指导下，在构建通用软件测试本体基础上，更要考虑军用装备软件测试的领域特点，构建包含军用装备软件测试的领域本体，以满足领域知识描述、构建和智能应用的需要。下面以某导引头装备软件测试实战中测试需求、测试项和测试用例为例子分析领域本体构建。

领域本体构建例 1：以某导引头装备的软件测试需求说明为例，可构建如下图所示的本体，其中不仅包含 ZZ 软件和功能等通用概念，还包含信号、中断信号等领域概念，以及优先级、中断周期、开启、清零、发送、接收等领域属性，并且包含两个领域公理：(1) 发送和接收的互逆关系；

(2) 开启和清零的属性上下位关系。因此，结合领域软件测试需求构造的本体具有明显的领域特点。

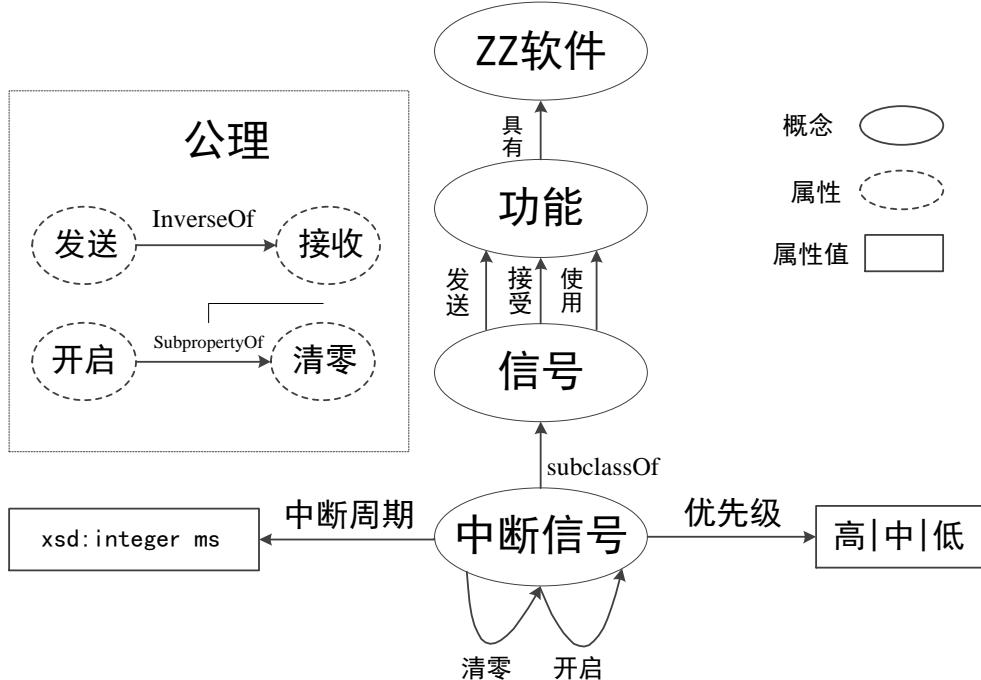


图 3-22 某导引头装备软件测试需求本体

领域本体构建例 2：根据某导引头装备的软件测试项数据，在领域专家协助下，整理出相关的领域概念和关系，并构建如下图所示的测试项本体，其中包含测试用例、设计人员、测试人员、测试监督人员、测试项、测试过程等通用软件测试领域概念，还包含参与、测试标识、测试追踪、测试说明、输入及操作说明、前提与约束等大量领域内关系，从而实现对该导引头装备软件测试项知识的规范描述。

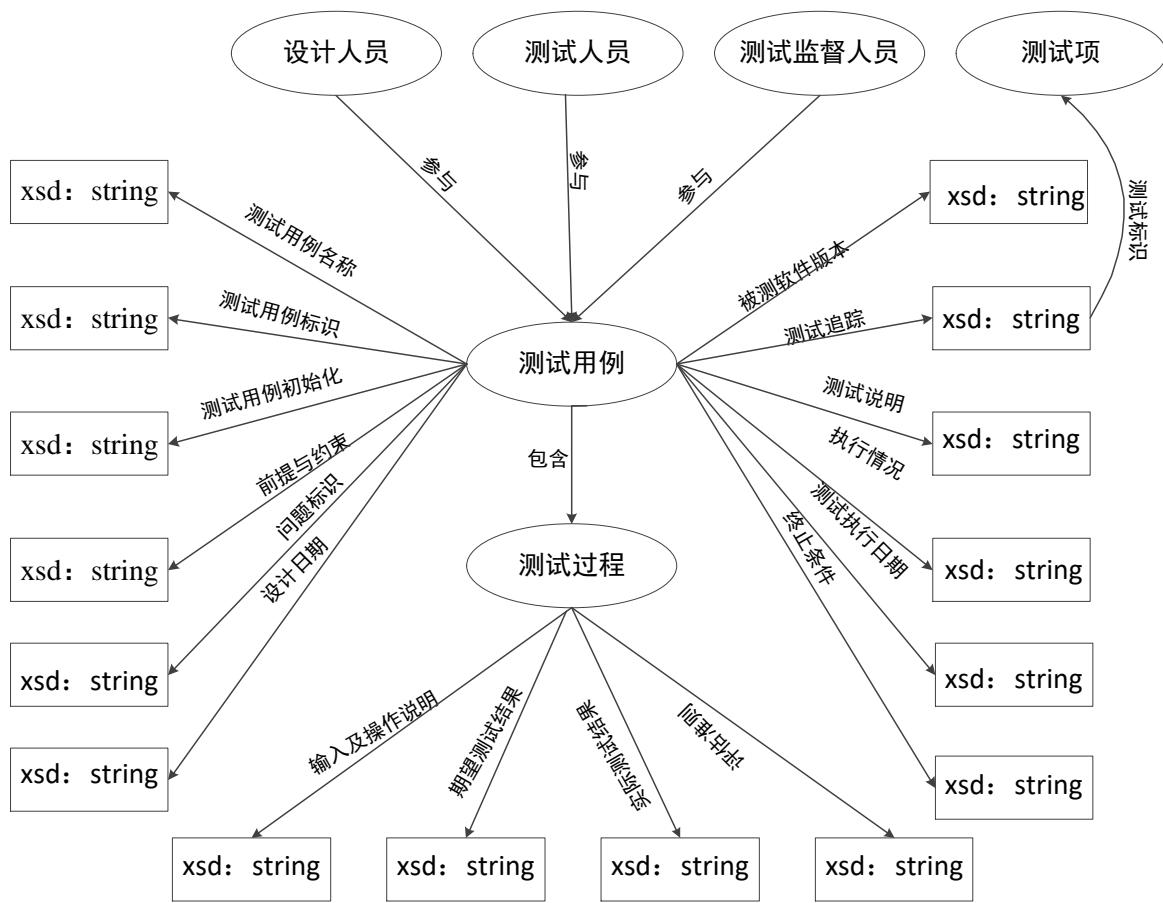


图 3-23 某导引头装备软件测试项本体

领域本体构建例 3：根据某导引头装备的软件测试用例数据，在领域专家的协助下，构建如下图所示的测试用例本体。该本体不仅包含研制总要求这种领域概念，还包含测试项、任务书、需求规格说明书、功能测试、性能测试、安全性测试、测试方法等通用软件测试领域概念，此外，本体中的关系有追踪和包含，属性有测试项名称、测试项标识等等。从该本体可见，测试用例本体和测试项本体具有一些语义等价或重叠的概念、关系或属性，可以将其进行合并，这也是本方案在后继知识融合中需要处理的任务。

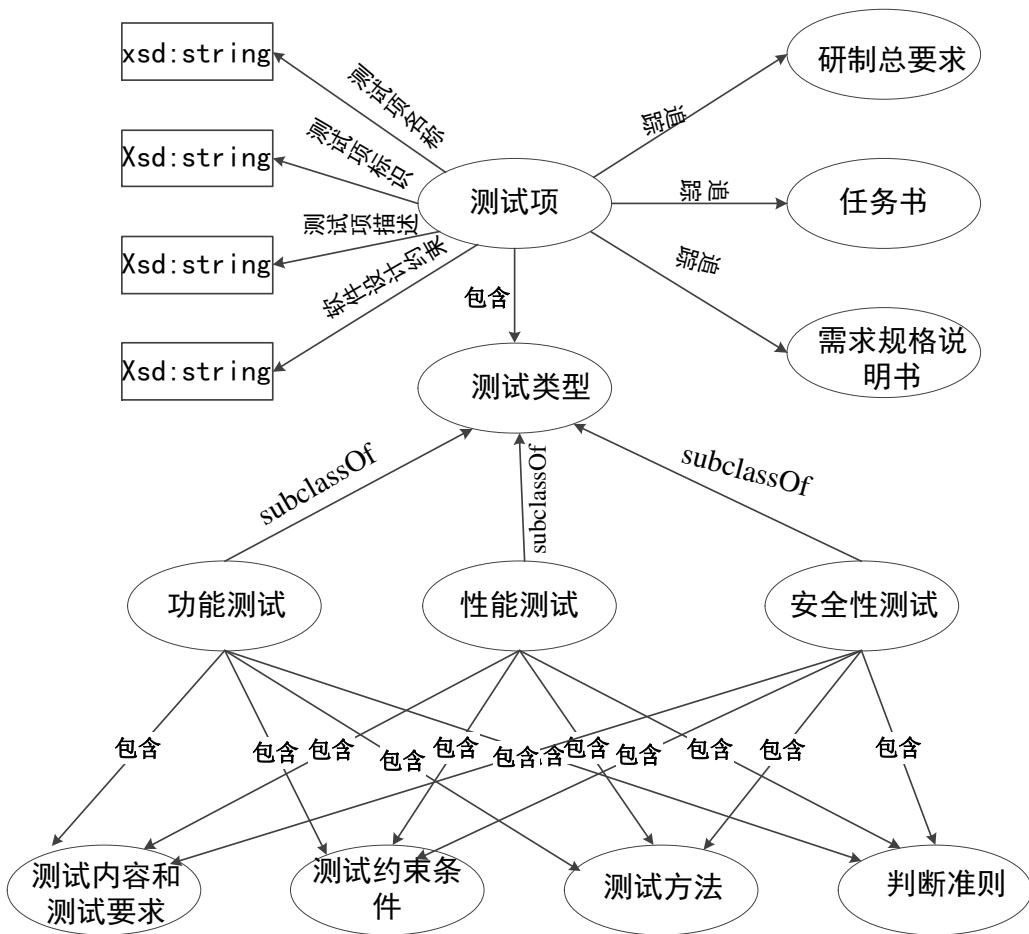


图 3-24 某导引头装备软件测试用例本体

综上所述，本方案在知识图谱构建各环节重点考虑领域知识的构建，即在军用装备软件测试中，大量测试任务和测试场景均需要考虑大量领域的知识，覆盖飞控、指控、测发控、雷达、导引头等典型装备软件类型涉及的知识，因此，本体构建和后继知识三元组抽取都必然要围绕军用装备软件测试领域设计针对性解决方案。

### 3.2 基于多策略的军用装备软件测试知识三元组抽取技术

本项目重点讨论在非结构化软件测试数据中如何有效地进行知识抽取，解决其中关键的实体识别、属性和关系抽取等问题。知识抽取是军用装备软件测试知识图谱构建的基础，针对该领域包含的结构化、半结构化和无结构化等多形态的数据，单一的抽取方法或模型不能完成知识抽取任务，因此，本方案采用多策略的抽取技术解决不同类型软件测试数据的知

识抽取。针对结构化和半结构化数据的知识抽取已有成熟的解决方法，软件测试中的代码相关信息也可以视为一种半结构化数据。实际应用中往往存在大量的无结构化文本数据，从这些文本数据中抽取高质量的知识一直是知识抽取的挑战性问题。面向非结构化文本数据的知识抽取是一个流水过程，抽取过程依次为：实体识别和关系抽取，分别从文本中获得实体和连接实体间的关系，最终得到高质量的知识图谱。通常而言，实体识别过程会对关系抽取结果造成直接影响，因为实体识别的效果直接影响了关系抽取效果。因此，在针对非结构化文本的知识抽取中，获取高质量的实体识别结果是一个重要的抽取目标。在军用装备软件测试领域的知识抽取中，还面临着训练数据集少和跨领域的知识抽取中数据标注工作量大等困难，采用远程监督可补充训练数据集，使用迁移学习可减少跨领域场景中的数据标注工作量。

### 3.2.1 面向结构化数据的测试知识抽取

软件测试过程的知识来源于支撑软件测试过程分析业务系统的结构化测试工件数据，因此，从数据库这种结构化数据中抽取知识也是一类重要的方法。在该领域，已经有一些标准和工具支持将数据库数据转化为 RDF 数据、OWL 本体等，转换过程包括 URI 的生成、RDF 类和属性的定义、空节点的处理、数据间关联关系的表达等，基本转换规则包括：

- (1) 数据库中的表映射为 RDF 类；
- (2) 数据库中表的列映射为 RDF 属性；
- (3) 数据库表中每一行映射为一个资源/实体，创建其 IRI；
- (4) 数据库表中每个单元格值映射为一个文字值(Literal Value)；
- (5) 如果单元格的值对应一个外键，则将其替换为外键值指向的资源或实体的 IRI。

本方案利用 R2RML 映射语言实现结构化数据到知识的转换，该语言是一种用于表示从关系型数据库到 RDF 数据集的自定义映射语言。这种映射

提供了在 RDF 数据模型下查看现有关系型数据的能力，并且可以基于用户自定义的结构和目标词汇表示原有的关系型数据。在数据库的直接映射中，生成的 RDF 图的结构直接反映了数据库的结构，目标 RDF 词汇直接反映数据库模式元素的名称，结构和目标词汇都不能改变。然而，通过使用 R2RML，用户可以在关系数据上灵活定制视图。每个 R2RML 映射都针对特定的数据库模式和目标词汇量身定制。R2RML 映射的输入是符合该模式的关系数据库，输出是采用目标词汇表中谓词和类型描述的 RDF 数据集。R2RML 映射是通过逻辑表（Logic Tables）从数据库中检索数据的。一个逻辑表可以是数据库中的一个表、一个视图或者是一个有效地 SQL 查询。每个逻辑表通过三元组映射（Triples Map）映射至 RDF 数据，而三元组映射是可以将逻辑表中每一行映射为若干 RDF 三元组的规则。“逻辑表”突破了关系数据库表的物理结构的限制，为不改变数据库原有的结构而灵活地按需生成 RDF 数据奠定了基础。

### 3.2.2 面向半结构化数据的测试知识抽取

半结构化软件测试数据具有较好的数据质量，是本项目中知识抽取的重要数据源，虽然它不符合关系数据库或其他形式的数据表形式结构，但包含标签或其他标记来分离语义元素并保持记录和数据字段的层次结构。一直以来，军用装备软件测试数据中的半结构化数据越来越丰富，全文文档和数据库不再是唯一的数据形式，项目组前期已积累的军用装备软件测试领域的大量半结构化数据，对于需求文档、测试规格说明和测试报告等测试产物中的半结构化数据，可以直接根据半结构化的语法进行数据解析，构建类似处理结构化数据知识抽取的规则，抽取出相关的知识。

针对军用装备软件测试中的代码片段、代码文件、代码文档、代码缺陷、代码开发者和测试者等重要的半结构化代码数据，抽取代码相关信息及其关联，形成军用装备软件测试知识图谱的重要组成部分，支撑智能化测试中代码缺陷预测、代码自动审查、代码语义搜索等智能应用。在软件

测试工件数据基础上，首先，抽取出代码相关的文档层的知识关联，即建立软件需求中关键算法与代码文件的关联，代码文件与测试需求文档的关联，代码文件与测试用例的关联；然后，抽取出代码相关的人和组织层的关联，即代码的编写者、需求制定人、测试人员、开发组、项目组间的关联；最后，抽取出代码层的知识关联，即代码包含的类、API 或方法、类间依赖关系、方法间调用关系，以及代码在历史测试数据中的特征。从而根据测试工件数据，提取代码相关知识，并建立知识间的关联。上述代码知识图谱的三层结构如下图所示。

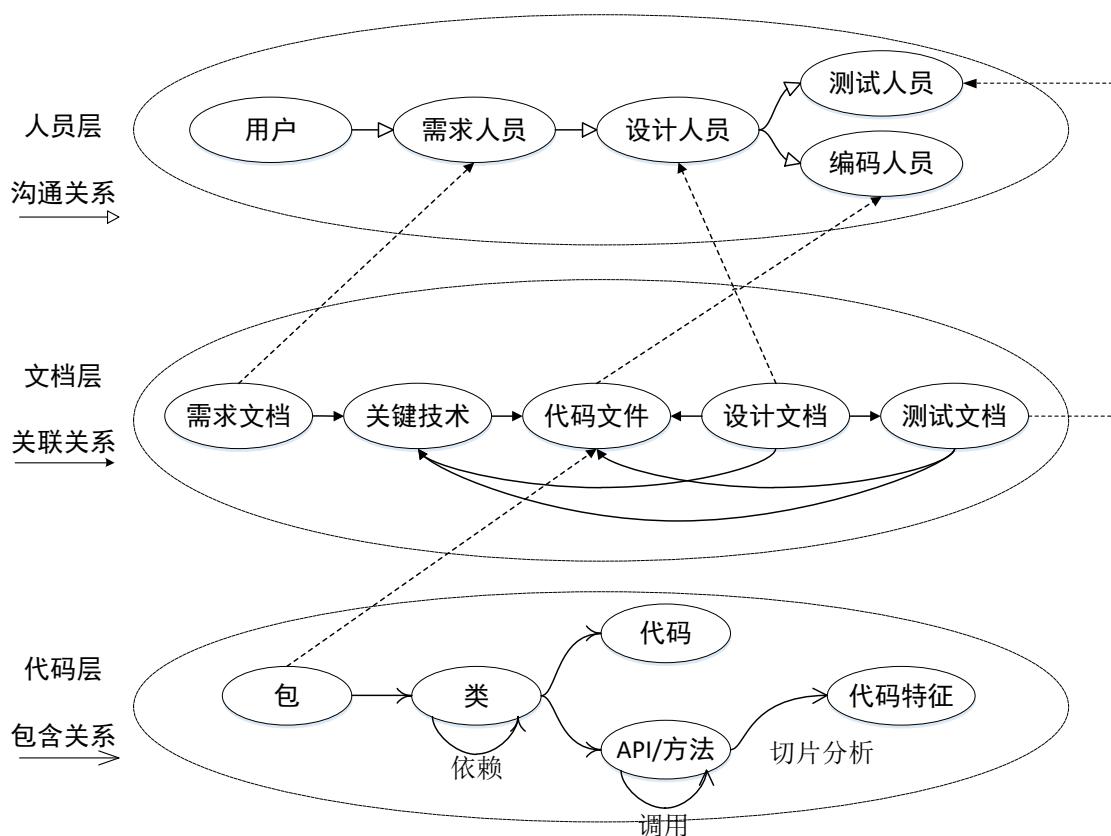


图 3-25 代码知识图谱的三层结构

### 3.2.3 面向非结构数据的测试实体识别

从软件测试相关的文本中进行命名实体识别是典型的序列标注任务，为了获得较好的复杂装备领域命名实体识别效果，本项目从四个方面来解决军用装备软件测试领域知识抽取的瓶颈：

首先，在传统的序列标注方法基础上引入和自然语言处理中的预训练过程，并通过深度学习得到的特征作为序列标注学习模型的输入。

其次，在训练过程中通过领域语料库进行训练，并利用领域词典，通过对抗学习技术来自动生成包含大量领域词典的新的训练文本，这些训练文本即包含领域词典内容，也具有领域语料库的特征，从而保证模型训练初始阶段拥有一定的训练数据。

再次，将命名实体识别和知识图谱构建视为一个自举迭代(Bootstrapping)的过程，利用现有军事装备领域词典、百科数据、已构建的知识图谱来对实体识别过程进行远程监督学习，生成训练数据集，抽取知识过程中，再使用已构建知识图谱，迭代进行远程监督生成测试数据，从而在解决军事装备领域软件测试构建数据集构造难题的同时，进一步提高命名实体识别的结果质量。

最后，针对军用装备软件测试中大量跨任务、跨应用、跨场景和跨领域软件测试中进行领域知识抽取的另一难题，即目标领域训练数据稀缺（标注稀缺和内容稀缺），数据标注工作费时耗力的问题，引入迁移学习技术，具体采用参数迁移和多任务学习迁移两种迁移方法，解决训练数据稀缺和标注工作量大的难题。

根据上述方案思路，在非结构化软件测试文本数据的实体识别中引入预训练模型。具体采用谷歌的 BERT 模型，它是现有最出色的基于 Transformer 思想的预训练模型，通过改造输入输出格式来适应下游自然语言处理任务。对于命名实体识别任务来说，首先对输入的句子加起始和终止符号，然后在输出部分 Transformer 的最后一层对每个单词进行分类即可。为了提高针对特定领域的命名识别效果，本方案引入领域语料库，可以实现对模型进行精细调谐，还可以对 BERT 进行扩展。针对命名实体识别任务，在 BERT 模型的预训练结果基础上，通过双向的 LSTM 深度学习模型获取命名实体序列标注特征，最后输入给 CRF 序列学习模型，即可获得命名

实体识别结果。双向 LSTM 和 CRF 模型已在大量的自然语言处理工作中证明对命名实体识别的效果，本项目提出的方法在此基础上引入了预训练和领域词典，能获得高质量的命名实体识别结果，随着知识图谱规模的扩大，在识别过程中利用知识图谱进行远程监督学习，形成了一个迭代的过程，可以进一步提高命名实体识别的效果。

在军用装备软件测试知识抽取中，面对一个新的装备型号、新的测试任务、新的测试场景时，由于难以有效获取符合要求的领域数据集，往往需要人工构造相应的数据集来满足模型训练要求，然而这是一个耗时费力的过程，难以适应新型武器装备的快速迭代和反复试验的研制需求，因此成为军用装备软件测试知识抽取的一个难点问题。为解决该问题，本方案引入了远程监督和自举迭代的技术，具体思路如下图所示。首先，通过项目组已积累的领域词典，包括军用装备领域词典、军用软件开发词典、军用软件测试词典等领域词典以及军事百科词条等开源数据，自动对软件测试工件的数据集进行标注；与此同时，利用已构建的军用装备软件测试知识图谱，对数据集进行自动标注。然后，为了保证标注数据结果的质量，通过投票机制过滤不可信的标注结果，并在后继模型训练中引入强化学习机制，奖励正确的标注，惩罚错误的标注，降低数据标签噪声，进一步提高标注结果质量。再次，利用自动标注和标注清洗的数据，训练知识抽取模型，抽取得到知识三元组。接着，将可信的新发现知识三元组加入到训练数据集中，通过自举迭代方式进行训练数据集的再次扩充和模型的训练，直到新发现的知识三元组较少为止。最后，得到的知识图谱同样可以再次对新的数据集进行标注，形成一个基于远程监督的测试数据自动标注、标注清洗、自举训练的迭代过程，从而有效解决领域数据集稀缺和数据标注工作量大的军用装备软件测试领域知识图谱抽取难题。

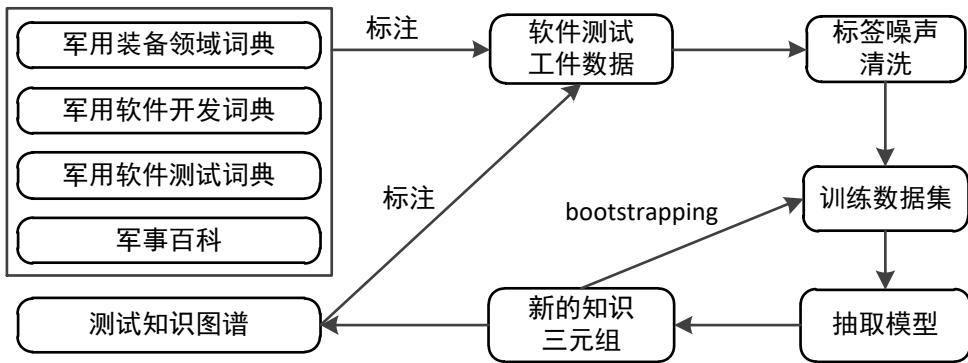


图 3-26 基于远程监督的军用装备软件测试领域知识抽取

针对以上军用装备软件测试知识抽取场景中面对跨型号、跨任务、跨场景测试中领域训练数据缺乏和数据稀疏的难题，还可以进一步引入迁移学习技术，在上述远程监督和自举迭代的基础上，提高跨任务、跨型号、跨场景的模型训练效率。迁移学习可利用语料资源丰富的已完成源领域中学到的软件测试知识，来辅助语料资源匮乏的目标领域上的学习任务。由于神经网络通常使用基于梯度下降的方法进行增量训练，因此直接在源领域和目标领域之间使用梯度信息进行优化以实现知识转移是可行的。根据源领域和目标领域中数据的关系，即特征空间、类别空间、边缘概率分布、条件概率分布的异同，本方案采用的基于神经网络的迁移学习方法包括参数初始化方法（INIT）和多任务学习方法（MULT），具体如下图所示。

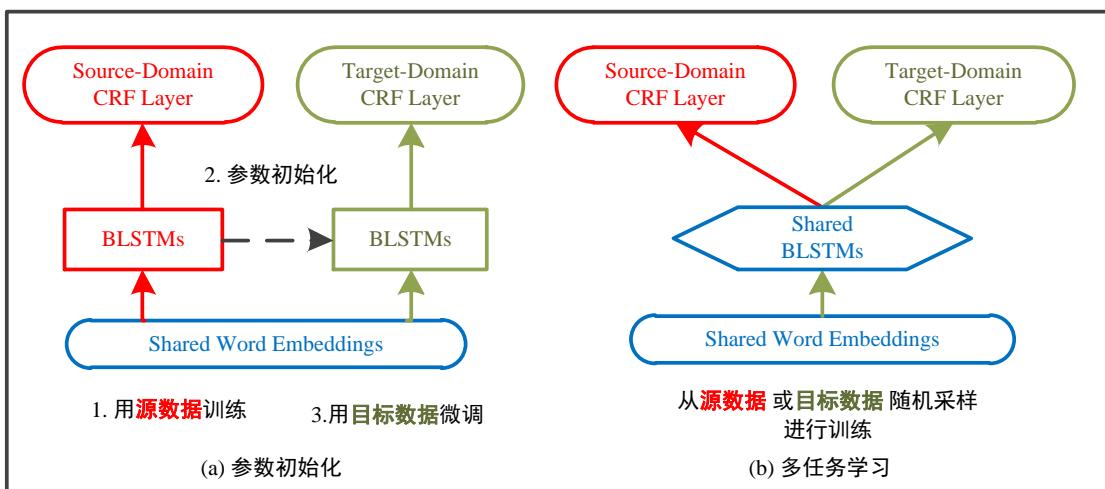


图 3-27 基于迁移学习的军用装备软件测试知识抽取

参数初始化迁移方法首先在源领域上训练神经网络模型，然后直接使用训练得到的模型参数对目标领域的神经网络模型进行初始化，从而实现从源领域到目标领域的迁移。迁移后，可以将参数固定，即不对目标领域的神经网络模型进行训练。但是，当目标领域仍有部分标注数据时，可以利用目标领域的标注数据对模型参数进行微调（Fine-tune）以得到更好的效果。多任务学习迁移方法同时使用源领域数据和目标领域数据对一个神经网络模型进行训练，按 $\lambda:1-\lambda$ 的比例分别对源领域和目标领域的数据进行采样，模型的损失函数为： $\mathcal{L} = \lambda\mathcal{L}_S + (1 - \lambda)\mathcal{L}_T$ 。其中， $\mathcal{L}_S$ 和 $\mathcal{L}_T$ 分别为模型在源领域和目标领域中的损失函数，且都经过了基于领域数据量大小的正则化， $\lambda \in (0, 1)$ 是为了平衡两个领域对最终结果的影响而引入的超参数。

将上述两种迁移方法运用在基于预训练的实体识别模型上，具体方法为：(1) 对于参数初始化迁移方法，首先，使用源领域训练数据对源领域模型进行训练，然后，用源领域模型中 LSTM 层的参数对目标领域的 LSTM 层进行初始化，之后可以选择两种策略利用目标领域的数据对目标领域模型进行训练：INIT-FineTune 和 INIT-Frozen，前者是对整个目标领域模型进行参数更新，后者只更新 CRF 层的参数。(2) 对于多任务学习迁移方法，首先，对源领域训练数据和目标领域训练数据进行随机采样，以超参数 $\lambda$ 为样本来自于源领域数据的概率，然后同时对源领域模型和目标领域模型进行训练，两个模型在训练过程中使用相同的 LSTM 层参数。这种方法在每构建一个新的目标领域模型时，都将重新训练源领域模型，当源领域训练数据规模较大时会产生较高的时间成本。

值得指出的是，本知识抽取方案思路不局限用于军用装备领域软件测试中的实体识别，还可以用于后继的软件测试关系抽取。

### 3.2.4 面向非结构数据的测试关系抽取

在面向非结构软件测试工件数据的测试关系抽取中，虽然可以针对特定场景人工定义关系模板，通过模板实现关系的抽取，但本项目进一步考虑能快速从大量多源的测试领域相关文本中学习新关系，即不在人工设定的关系抽取模板中的关系。本项目将基于卷积神经网络的关系抽取模型扩展到远程监督数据上，其原理是假设每个实体对的所有句子中至少存在一个句子反应实体对的关系，则对于每个实体可以通过大量的训练学习到最能反应其关系的那个句子。针对远程监督方法的噪音问题，进一步引入基于句子级别注意力机制的神经网络模型来解决这个问题，该方法能够根据特定关系为实体对的每个句子分配权重，通过不断学习能够使有效句子获得较高的权重，而有噪音的句子获得较小的权重，从而可以提高这种关系抽取的效果。解决远程监督存在的噪音的另外一种思路是通过强化学习技术，对噪音低的学习进行奖励，而对高噪音的学习进行惩罚，从而实现远程监督中数据降噪问题。对于两种降噪的处理，在研究过程中将通过实验对比来进行方法选型。

基于上述多策略的军用装备软件测试知识三元组抽取，针对不同形态的军用装备软件测试工件数据，采用相应的抽取策略，获取数据中蕴含的知识，实现从数据到信息，从信息到知识的转换过程。下面以某导引头装备软件测试实战中测试需求、测试项和测试用例等数据中的知识三元组为例进行分析。

领域知识图谱例 1：根据上述知识抽取方案，针对某导引头装备的软件测试需求文本，根据 3.1 中构建的导引头装备软件需求本体，综合采用上述实体识别和关系抽取方法，得到相应的需求知识图谱，如下图所示，图中左边是原始需求文本，右边是本方案模型抽取的知识图谱。该知识图谱中，功能包含上位机、下位机和中断控制等实体，信号包括中断 4、中断 6、中断 7 等实体，同时实体的属性值，例如中断 4 的优先级为高，以

及实体间的关系，例如中断 4 可开启中断 7，均被抽取出来，并形成直观的知识图谱，形成对后继测试智能服务的支持。

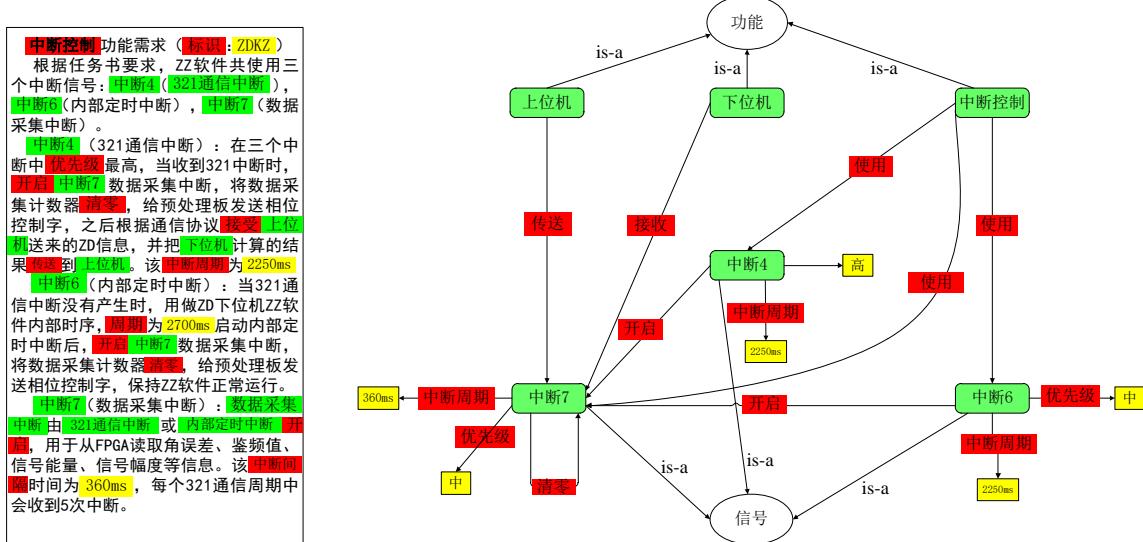


图 3-28 某导引头装备软件测试的需求知识图谱

领域知识图谱例 2：根据某导引头装备软件测试的测试项本体和测试项数据，基于本方案抽取得到其中的测试项知识图谱，如下图所示，(a) 是测试项原始数据，(b) 是对应的测试项知识图谱。其中的核心实例是“中断控制测试”，它包含“功能测试 A”、“性能测试 A”、“安全性测试 A”等测试实例，以“功能测试 A”为例，其又包含“测试内容和测试要求 A”、“测试约束条件 A”、“测试方法 A”、“判断准则 A”等实例，每个实例均有具体描述内容。当然，“中断控制测试”也包含了测试项描述、测试项名称、测试项标识等典型的属性值，在“中断控制任务书 A”中，还包含“4.1.2 控制”等实体。该知识图谱与相应的测试项本体和测试项半结构化数据对应。

测试项名称	中断控制	测试项标识	ZDKZ
追踪关系	研制总要求: 四(九) ZD下位机ZZ软件的功能、四(四) 主要性能指标和使用要求 任务书: 4.1.2 控制、4.2 性能、4.4.3 DSP中断 需求规格说明: 3.2.2 中断控制、3.2.21 性能、3.2.22 资源裕量		
测试项描述	考察程序三个中断的执行.....		
软件设计约束	无		
测试类型	功能测试	1测试内容及测试要求: 每2000ms能正确响应一次321通信中断, ..... 2测试约束条件: 正确连接各测试设备和通信接口模拟器。 3测试方法: 使用插桩的方法..... 4判断准则: 321通信中断时间间隔为2250.....。	
	性能测试	1测试内容及测试要求: 数据采集时间应小于2700ms, ..... 2测试约束条件: 正确连接各测试设备..... 3测试方法: 通过上位机模拟器..... 4判断准则: 每次采集的计时结果均应满足.....。	
	安全性测试	1测试内容及测试要求: 下位机321通信有误, ..... 2测试约束条件: 正确连接各测试设备..... 3测试方法: 在程序定时器1中断..... 4判断准则: 当断开321通信后, .....	

图 3-29 (a) 某导引头装备软件测试的测试项原始数据



图 3-29 (b) 某导引头装备软件测试的测试项知识图谱

领域知识图谱例 3：根据某导引头装备软件测试的测试用例本体和测试用例数据，基于本方案抽取得其中的测试用例知识图谱，如下图所示，(a) 是测试用例原始数据，(b) 是测试用例知识图谱。在该知识图谱中，包含了一个测试用例实例“测试用例 A”，它又包含“测试介绍 A”、“测试记录 A”、“测试过程 A”等实例。以“测试记录 A”为例，其又包含设计人员“王武”、问题标识“13”、测试监督员“李思”。值得注意的是，本方案可从“测试过程 A”中包含的输入及操作说明和期望测试结果的属性值文本中，识别一个方法实体：Timer\_350ms()，基于其它相关的测试工件代码数据，在知识图谱中可进一步抽取该方法所属类的实体 IntController，以及对应的代码特征。

被测软件版本	4.A.00			
测试用例名称	中断控制-内部定时器1			
测试用例标识	ZDKZ-GN-001			
测试追踪	ZDKZ			
测试说明	正确响应内部定时器1中断			
测试用例 初始化	下位机控制台“加电”，伺服“加电”，打开信号模拟器			
前提与约束	测试用各设备工作正常，测试仪器在有效期内，正确连接各测试设备和通信接口模拟器			
终止条件	测试用例正常终止：该测试用例按正常测试步骤完成测试过程；对于因测试环境限制无法执行的测试用例已采用其它手段验证了相关需求的实现。 测试用例异常终止：测试环境故障、测试前提条件不具备；发现重大问题。			
测试过程				
序号	输入及操作说明	期望测试结果	评估准则	实际测试结果
1	修改程序，在timer_25ms() 模块设置断点，然后启动程 序，正常运行	程序正确进入 timer_25ms() 模块	与期望结果一致	
设计人员	王武		设计日期	20131110
执行情况		执行结果		问题标识
测试人员	张伞	测试监督员	李思	测试执行日期

图 3-30 (a) 某导引头装备软件测试的测试用例原始数据

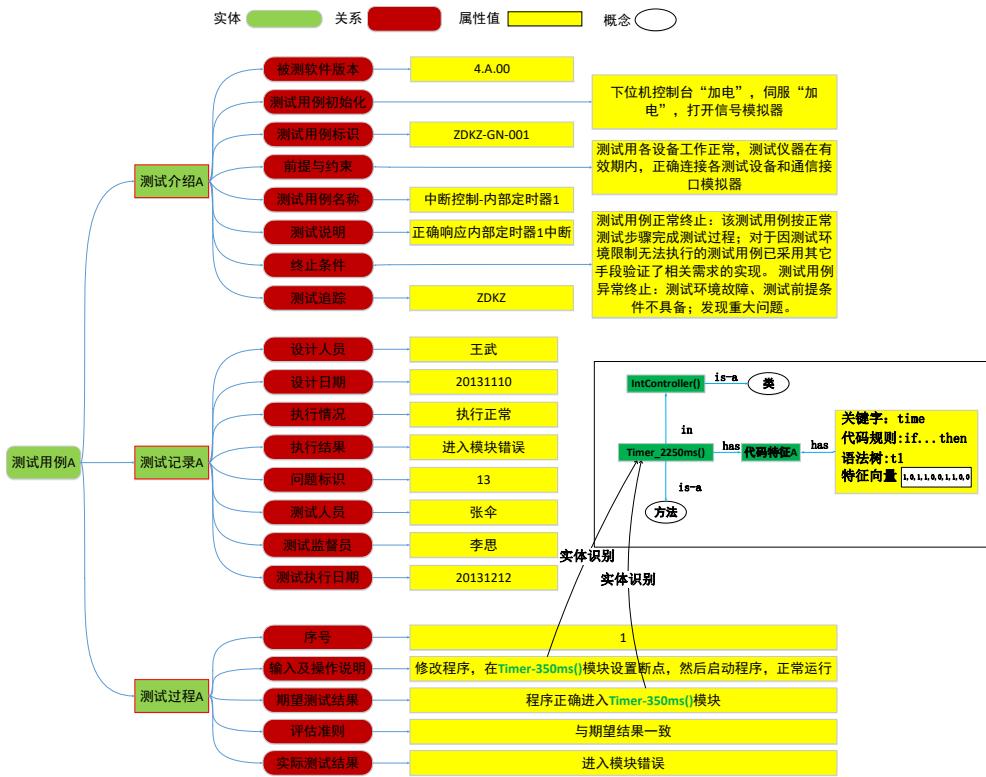


图 3-30 (b) 某导引头装备软件测试的测试用例知识图谱

综上所述，本方案针对不同形式的军用装备软件测试数据，采用不同抽取策略实现其中知识三元组的抽取，并在此过程中引入领域词典、预训练模型、远程监督和迁移学习等前沿技术，提高知识三元组抽取的质量和性能，并支持飞控、指控、测发控、雷达、导引头等典型装备软件类型的领域知识抽取。

### 3.3 基于语义子图和约简锚点的大规模测试知识图谱融合技术

在基于多源异构数据的军用装备软件测试知识图谱构建中，由于数据源的分布和自治性，在知识图谱构建中往往存在大量的异构问题，主要包括本体层的异构和实例层的异构两个方面。例如，在前述某导引头装备软件知识图谱中，本体中的两个关系“中断周期”和“中断间隔”是匹配的属性，实体中的“中断 4”和“321 通信中断”、“中断 6”和“内部定时中断”、“中断 7”和“数据采集中断”均是等价的实体，应将其进行融合。本知识融合方案拟通过本体匹配和实例匹配，解决这本体层和实例层两方面的知识异构问题，实现大规模知识图谱构建过程中的高效融合。

### 3.3.1 测试知识图谱本体层融合

知识图谱本体层融合，通常通过本体匹配的方式得以实现，在基本的语义匹配基础上，本项目拟提出一种基于语义子图的软件测试本体匹配，能有效解决软件测试知识图谱的本体层融合问题，该匹配方法具有适应性广，匹配精度和召回率高的特点。尤其值得指出的是，软件测试数据很多情况下具有弱信息的特点，即其中供人阅读的文本信息有限，包含大量的编码等，而这里提出的基于语义子图的本体匹配技术正是擅长解决这类问题。其基本思路如下：

给定两个异构本体，对其进行匹配预处理，预处理工作主要包括解析本体和准备某些匹配算法所需的数据结构，在本研究中，我们在预处理阶段构造语义子图，用于对本体中元素语义的精确描述。利用预处理阶段获得的语义子图，采用基于语义子图的文本匹配方法为弱信息本体对生成少量高质量的匹配结果，并将其作为相似度传播的初始种子。具体方法如下图所示。

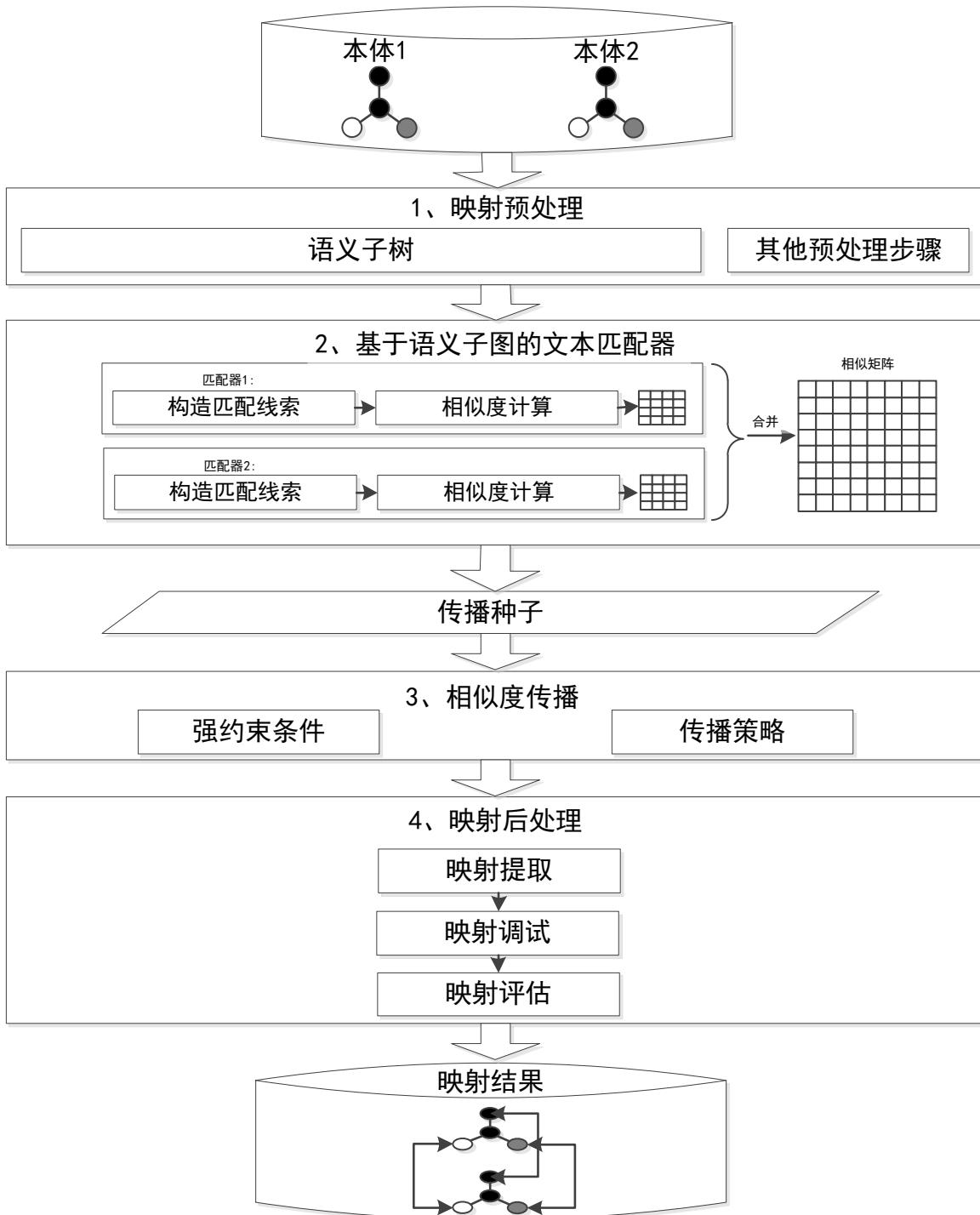


图 3-31 基于语义子图的本体匹配

采用强约束条件相似度传播模型对若信息本体进行结构匹配，传播过程以初始匹配种子为输入，并在传播的不同阶段采用合适的传播策略。传播过程迭代进行，直到满足传播结束条件为止。经过相似度传播，将得到

一个相似度矩阵，再通过匹配提取和匹配调试技术，就能得到最终的匹配结果。

本课题提出一种基于语义子图的文本匹配器来为相似度传播模型提供高质量的初始种子。首先将本体元素的相关文本按照其对元素语义的描述方式进行组织和划分，并称由此构造出的文本为元素的语义描述文档(Semantic Description Document)。为了避免引入与元素关联不大的其它文本，语义描述文档构造过程被严格限制在元素的语义子图内，即语义描述文档是基于语义子图的。此外，语义描述文档不考虑本体语言中的元语，如`rdfs:Class`和`owl:hasValue`等。构造过程还包括文本预处理，即进行词干提取和频繁词去除等。

本课题希望寻找到合理的方式来组织概念和关系的相关文本，并利用这些文本来作为判断匹配的线索。为达到这个目的，这里从两个方面进行考虑：一方面，根据概念和关系的语义描述方式对文本进行重新组织，这种组织方式具有通用性；另一方面，为了避免得到的文本中包含那些与元素并不太相关以及不太重要的文本信息，将描述文本构造的范围局限于语义子图内。

构造概念和关系的语义描述文档后，便可通过计算语义描述文档相似度来寻找异构本体元素间的匹配。两元素的语义描述文档相似度越高，它们相匹配的可能性越大。描述文档根据本体对元素描述的语义特点被划分为不同的类型，所以相似度计算是在相同类型的文档中进行的。

### 3.3.2 测试知识图谱实例层融合

在面向复杂装备软件测试知识图谱的实例层融合中，大规模知识图谱中的实例匹配是一个核心的挑战问题，针对该挑战，本项目提出一种解决大规模知识图谱中的实例匹配解决方案。首先，利用图数据库技术实现大规模实例或链接数据的存储，在此基础上构造针对三元组、概念、属性、实例的RDF声明哈希索引，并对具有相似的实体集合建立局部敏感哈希索

引，这不仅解决了大规模实例匹配的空间复杂度，并为后继匹配计算提供了高效的常用数据解析操作；然后，利用经典的基础匹配器完成匹配相似度计算，并根据匹配任务设计选择合适的匹配方法；再次，为解决匹配计算这一最大的性能瓶颈，将数据分块和匹配锚点预测两种最有效的高效技术进行综合，并进一步采用分布式计算模式来完成匹配计算；最后，匹配调试可去除不一致结果，匹配选择输出 1:1 的匹配结果，还可以利用少量匹配结果作为样本，对系统参数进行调优。具体思路如下图所示。

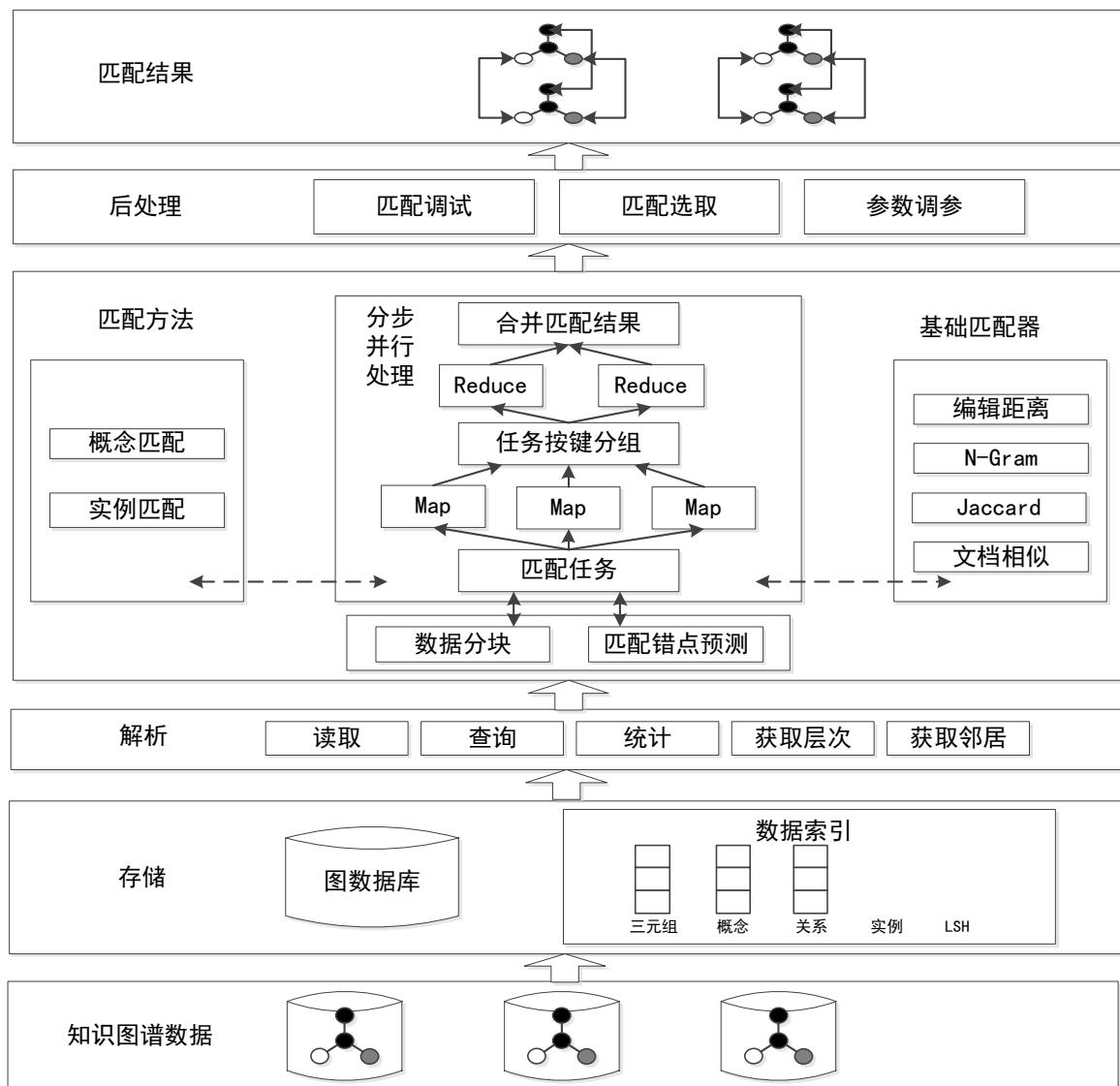


图 3-32 大规模知识图谱实例层匹配技术架构

根据上述研究总体思路，并结合国内外已有研究和申请团队在知识融合方面的研究基础，本项目拟对大规模知识图谱中实例匹配面临的三方面的核心问题进行系统深入研究，具体工作如下：

### （1）匹配算法的空间复杂度处理

大规模实例匹配的空间复杂度有两个来源：读入和解析知识图谱需要的存储空间；匹配计算过程中的各种数据结构存储空间。解决第一种空间复杂度需要研究如何通用多次读取或利用数据库辅助存储实例等技术。对于第二种空间复杂度，除了需要考虑数据结构的设计，并利用数据压缩技术节约存储空间外，还需要考虑在具体系统实现中及时对无用的垃圾变量进行释放，最大可能降低匹配计算的辅助空间。

### （2）匹配算法的时间复杂度处理

本课题将探索一种新颖的分治策略来降低大规模软件测试领域实例匹配的时间复杂度，该分治策略具有通用性，适用于多数匹配器。我们注意到大规模实例匹配中普遍存在两种事实：大规模软件测试知识图谱包含大量由 *is-a* 和 *part-of* 关系构成的层次结构，正确的匹配不能破坏这种层次结构。大规模实例间的元素匹配具有区域性特点，即知识图谱  $O_1$  的特定区域  $D_i$  中的元素大多会被匹配到知识图谱  $O_2$  的特定区域  $D_j$  中。本课题将研究如何从上述两种事实出发提出新的大规模实例匹配的分治策略，该策略应能避免传统分治策略对大规模实例的划分处理，具有通用性。

### （3）通用的大规模实例匹配框架

在上述研究的基础上，本课题将提出一种通用的大规模实例匹配框架，该框架将大规模实例匹配处理划分为几个独立的主要模块，即预处理模块、分治策略实施模块、匹配器选择模块和匹配后处理模块等。针对不同的模块，既便于分别解决其中的空间复杂度和时间复杂度问题，也便于吸收各种实例匹配的最新研究成果用于解决大规模实例匹配难题。

### 3.4 基于最小代价的知识动态更新

本方案针对动态变化的军用装备软件测试知识，开展知识演化操作和知识演化策略研究，通过知识演化操作定位知识图谱演化位置，根据知识演化影响力分析，研究最小代价知识演化方法，解决软件测试知识图谱演化中的知识更新代价问题。

本方案认为，软件测试流程知识的一个完整的演化流程包括六个阶段：变化捕捉（Capturing）、变化表示（Representation）、语义变化（Semantics of change）、变化实施（Implementation）、变化的传播（Propagation）和变化的确认（Validation），以此往复，维持知识的运行。根据这一流程可以得出，变化的捕捉和变化的表示是实现知识演化的基础。

变化的捕捉就是通过显式的需求或变化的自发现方法来捕捉知识的变化。一般在用户修改知识时记录下对知识的变化操作，生成变化日志。而在本项目中要在缺乏日志的情况下，捕捉变化。变化通常由领域内部或者外在的需求变化引起，这样我们可以从明确的需求中或者一些特定方法来捕获变化。捕捉变化，为知识撤销、修正变化或修复之前的知识提供了有力的支持。变化的捕捉类似于知识的匹配又不同于知识的匹配。知识匹配是指将知识中的概念相互比较，建立它们之间的映射关系。而变化的捕捉则是判断概念等元素是否完全相同。因此，我们可以利用知识匹配的一些思想，但不能完全照搬。一般在用户修改知识时记录下对知识的变化操作，生成变化日志。而我们的算法要在缺乏日志的情况下，发现这些变化。

在知识的局部，比如概念、属性、实例等知识元素部分发生了改变，即为知识元素发生了变化。所有局部的变化组合起来，即为完整的知识的演化。要想完整全面的发现知识的演化，就必须从捕捉知识元素的变化入手，通过捕捉不同元素的变化继而才能完整地发现知识的演化。

捕捉知识的变化后，还需要对变化进行定义和划分。定义越丰富，划分越详细，最终显示的变化信息定位也越详细，这也就为后续的知识演化

的实现以及确定知识演化的影响等工作提供了更丰富的参考信息。但同时意味着需要提供的变化处理操作也越复杂，算法的繁琐度也越大。

对于知识图谱，其演化过程中所执行的演化策略或演化路径所耗费的代价为：

$$C_{st} = \sum_{i=1}^j n_{Ch_i} \times C_{Ch_i}$$

式中 $st$ 代表演化中所选择的演化策略， $Ch_i$ 代表演化策略中应用到的变化操作， $n_{Ch_i}$ 就是变化操作 $Ch_i$ 的数量， $C_{Ch_i}$ 代表相应变化操作的代价因子。最小代价演化策略的核心是在决策点的决策选择以及在此决策点上的代价计算，当知识图谱在一个决策点后面临若干决策选项时，该策略就开始分析计算每一个策略以及后续所需耗费的代价，然后选择最小的代价，这也是最小代价演化策略名称的由来。同时值得注意的是，在一个决策点去选择代价最小的决策算法，并不一定能保证最终得到的演化策略或是演化路径的总代价就是最小，因为较小的代价的变化操作可能会导致操作数量的不断增多，从而造成决策点的增多最终代价随之增大。因此我们的演化策略算法在每一个决策点会递归调用自身直到变化操作所代表的决策点没有新的衍生操作。

本方案将知识的变化操作分为两个层次，第一层次为基本变化操作，第二层次为复合变化操作。基本变化操作是指只更改知识模型中的一个实体，它不能再分解为更简单的变化。复合变化操作指的是对知识中多个实体进行变更操作或者对一个实体进行多次变更操作。

基本稳定度从知识图谱元素匹配数目的角度提供了一种量化知识图谱演化影响的基本方法，然而这种度量方法是存在一定局限性的，同时也够全面。知识图谱是一种特殊的语义模型，知识图谱的演化也可以看做某种程度上知识图谱对语义的变化做出的反应。而基本稳定度仅仅从元素简单匹配的角度考虑问题，而忽略了知识图谱演化过程中的语义因素，因

此在这里我们提出一种语义稳定度的度量方法,改进基本稳定度中的一些不足。

对于一个知识图谱元素一般存在三种基本变化形式,分别是删除、增加和修改,这三种变化在知识图谱演化前后的版本中所对应的语义影响是不一样的,这里需要综合考虑这三种基本的变化操作。

对于删除和增加来说,由于是完全更换了内容,因此无论是旧有的被删除的元素还是新增加的元素都与相对应版本知识图谱中的元素失去了语义联系,而对于被修改的元素而言,由于其是由原版本知识图谱中的元素变化而来,因此存在着一定的语义联系,在本项目中,这种语义联系可以用文本相似度的值来度量。

综合考虑所有存在语义联系的元素,语义稳定度可以用下面的公式来表示:

定义 语义稳定度:

$$stab_{semantic} = \frac{2 \times \left( |E_i \cap E_j| + \sum_1^{|E_{modify}|} sim \right)}{|E_i \cap E_j|}$$

式中 $|E_{modify}|$ 表示在演化过程中被修改元素的数目, $sim$ 表示各被修改的元素与原知识图谱中对应元素的文本相似度值。

同样根据这一表述,我们可以利用语义稳定度来得出知识图谱演化的语义影响度量。

### 3.5 知识图谱质量评估

由于软件测试是软件质量的保障,本项目构建的软件测试知识图谱是智能软件测试技术的质量保障,所以务必要对软件测试知识图谱进行质量评估。对于知识图谱的评估,学术界和工业界没有一个公认的标准,因此本项目将研究软件测试知识图谱质量评估指标,并构建软件测试知识图谱质量评估体系。

#### 3.5.1 软件测试知识图谱质量评估维度

本项目认为软件测试知识图谱的生命周期可划分为三个阶段：产生阶段、使用阶段和维护阶段。根据每个阶段不同的特点，本方案将在每个阶段选择不同的质量维度对软件测试知识图谱进行质量评估，例如在生产阶段将需要的质量维度有三类：可信性、表示性和冗余性；在使用阶段将需要的质量维度也有三类：实用性、外在特性和内在特性；在维护阶段将需要的质量维度有一类：资源动态性。

本方案从可信性、表示性和冗余性三类维度对知识图谱生产阶段的质量进行评估。可信性是指数据的公信力，它通常包含了4个质量维度，分别为：出处、信誉、可信度、可验证性。但是本项目知识图谱是专业领域的知识图谱，构建图谱的数据是来自于任务书、设计文档等一些可信度较高的软件测试相关数据源。所以本方案认为不需要考虑信誉、可信度、可验证性这三种维度，对于可信性的评估只需要简单考虑数据出处，比如来源于哪个文档，文档的发布者是谁，由谁进行了改动。表示性维度是指数据提供者如何描述数据相关的维度，包括可理解性和可解释性。

冗余性主要是指数据集没有无关的、多余的信息，包括的质量维度有简洁性和语义冗余。在对简洁性的研究中，一般都将数据简洁性分为模式层和数据层。在模式层，数据简洁性指不包含冗余的属性，即用不同的名称代表相同的属性。因此，模式层的简洁性与唯一属性的个数相关，可以通过计算唯一属性的个数占所有属性个数的比例来测量。在数据层，数据简洁性指不包含冗余的对象，即用不同的标识去描述相同的对象。因此，数据层的简洁性与唯一对象的个数相关，可以通过计算唯一对象的个数占所有对象个数的比例来测量。数据简洁性与属性、对象的唯一性相关，也与数据相关性关联。

RDF 数据集的语义冗余是指如果将其中的一些三元组删除并不会导致数据含义的改变；从更广泛的角度来说，语义冗余指三元组模式的共同出现。在大部分情况下，删除这些三元组需要添加新的规则到数据集中，

这样在需要的时候可以重新生成这些被删除的三元组。对于语义冗余，可以通过定义图模式来确定规则去除冗余的三元组。

在软件测试知识图谱使用阶段，本项目使用实用性、外在特性和内在特性对使用阶段的质量进行评估。实用性是从用户使用的角度来评价的质量。包括可查询性和信息性两个维度。对于可查询性，主要通过构建查询的时间、尝试的次数两个客观指标和一个查询难度主观指标来评价。信息性则直接采用准确率、召回率、返回信息的完整性等常用客观性指标及一个信息率的主观指标来评价。数据的可访问性指数据是否可以正常的被访问和检索，包括可用性、性能、安全性和响应时间这 4 个维度。内在特性维度与数据内容相关，体现数据的正确性、逻辑一致性等。内在特性包括了 5 个维度：完整性、一致性、准确性、客观性和连通性。

资源动态性与数据发布后期对数据的维护相关，即随着时间的推移，使数据集保持动态性，适应真实世界的变化。由于本项目所研究的知识图谱具有可扩展性和自学习性，所以对于资源的动态管理有着一定的要求。

### 3.5.2 软件测试知识图谱质量评估方法

在上述质量评估维度中，有一些是定性的评估，有一些是定量的评估。对于定性的评估可以采用人为定义的标准进行打分，对于定量的评估可以使用一种简单的评分方法，比如对于语义准确性维度的评估可以将正确数据的数量除以总数据数量得到一个正确数据的比例，这个结果可以直接作为评分结果。由于本项目知识图谱有不少于几十万的实体和关系，所以很难对所有的实体和关系进行验证。为了保证数据质量评估结果准确的同时尽可能的减小成本，有必要对数据进行抽样评估。由于本项目知识图谱是会不断更新的，所以需要考虑动态的抽样策略。本方案使用一种基于分层抽样的增量评估方法。

在聚类采样过程中使用两阶段加权聚类采样，在第一阶段使用加权聚类采样在全部聚类中采样部分聚类，在第二阶段中在每个聚类中再进行三

元组的采样。通过这样两阶段的采样可以尽可能节省成本，非常适用于本项目的大型知识图谱。抽样后的三元组可以通过众包等人工评估方法进行评估。

增量评估方法的思想是基于知识图谱的更新是成批的，可以很自然的将每一个更新批次视为一个层次。也就是当知识图谱从 $G$ 演化到 $G + \Delta$ 时， $G$ 和 $\Delta$ 是两个独立的、互不重叠的层。分层抽样可以将 $G$ 和 $\Delta$ 的估计结果结合起来，计算出 $G + \Delta$ 总体质量评估结果的无偏估计。

#### 4. 知识快速检索与智能推送及测试用例生成技术

知识快速检索与智能推送及测试用例生成技术总体方案如下图所示，总体方案体系结构由知识图谱语义索引技术、基于知识图谱的测试工件智能推送技术和基于知识图谱的测试用例自动生成技术组成。

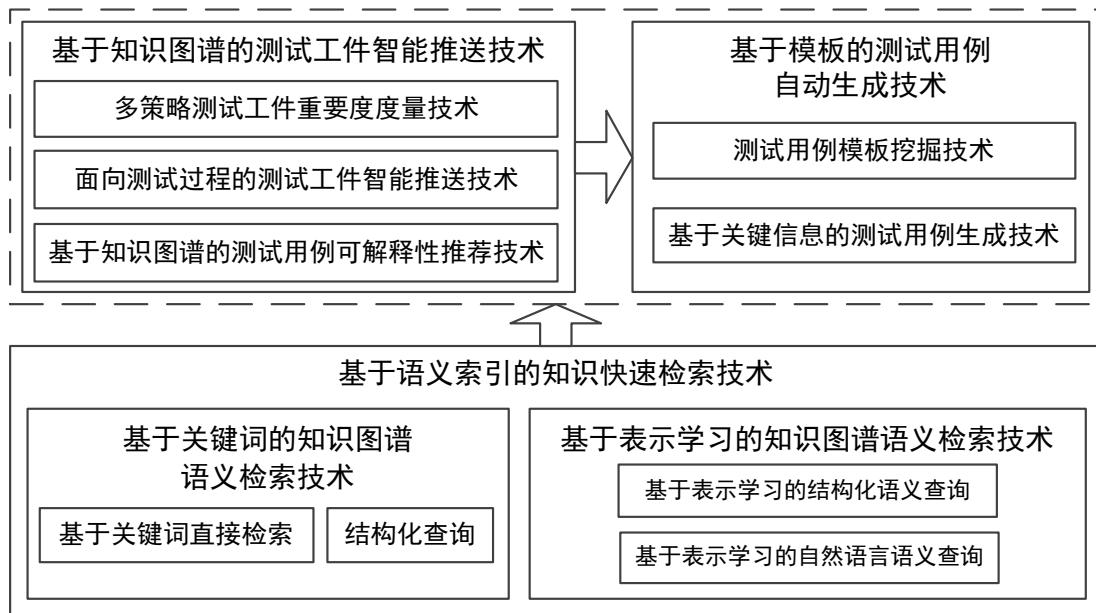


图 3-33 知识快速检索与智能推送及测试用例生成技术总体方案

首先，知识图谱语义检索技术结合关键词检索和知识表示学习实现测试知识语义检索。基于关键词生成结构化的查询，实现在知识图谱上直接检索知识；此外，通过知识图谱表示学习技术，将知识节点投射到低维连续的向量空间中，从而将相似知识映射到相邻空间内，以实现知识快速检索。

然后,基于知识图谱的测试工件智能推送技术依据知识图谱中存储的不同测试阶段产生的测试工件,制定不同阶段的推送策略和方式,结合知识语义索引技术,实现测试工件智能化推送,提升评测效率与质量。

最后,基于模板的测试用例自动生成技术挖掘测试项和测试用例中的关键命名实体,分析测试项和测试用例中的槽位置,形成测试项-测试用例模板。然后依据测试项模板分析获取新测试项中的关键信息,然后基于关键信息实现基于模板的测试用例生成。

#### 4.1 软件测试知识图谱语义检索技术

从软件测试知识图谱中检索知识,辅助测试人员进行测试设计是装备软件智能化测试的核心。武器装备软件具有严格的规范要求,往往要求检索系统具有很高的查准率。然而,传统的检索系统仅仅考虑关键词信息,遗漏了测试工件等知识间的关联关系,导致查准率下降,难以满足武器装备软件要求。软件测试知识图谱将从测试过程数据中抽取的知识进行关联,蕴含了测试知识间丰富的语义信息,在软件测试知识图谱上进行测试知识语义检索可大大提高软件检索的准确率,为后续测试工件推荐奠定基础。

因此,本项目研究软件测试知识图谱语义检索技术,总体方案如下图所示,总体方案的体系结构由基于关键词的知识图谱语义检索技术、基于表示学习的知识图谱语义检索技术组成。

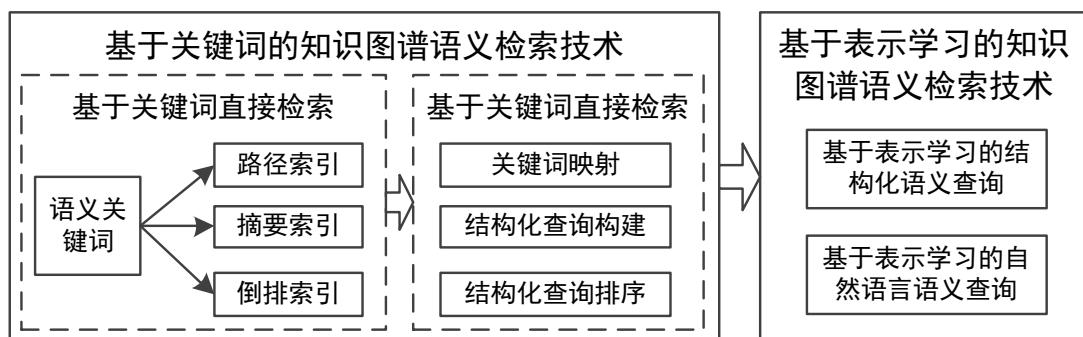


图 3-34 基于语义索引的知识快速检索技术总体方案

知识图谱检索是将用户的查询输入构建成结构化的查询语言 SPARQL,或者让用户直接提出结构化的查询,然而这种方式需要用户熟悉知识图谱

数据源、知识图谱的数据模式。然而，普通用户通常输入的是自然语言文本，难以直接进行知识图谱检索。所以，知识图谱的有效语义搜索需要一种简单高效的搜索范式，即允许用户以直观的、透明的、易用的方式对数据进行查询和浏览。基于此，本项目拟引入基于关键词的知识图谱语义检索技术和基于表示学习的知识图谱语义索引技术，实现知识图谱语义检索。

首先，基于关键词的知识图谱语义检索技术采用分词、句法分析、命名实体识别等手段获取问题查询语句核心关键词信息，然后基于关键词生成结构化的查询，结合知识图谱各节点的本体语义概念，实现知识语义检索。

然后，基于表示学习的知识图谱语义索引技术旨在通过机器学习技术，将知识图谱中的实体和关系投射到连续低维的向量空间中，将实体相似度计算转化为数值型计算，可快速对语义查询进行拓展，进而采用基于表示学习的结构化语义查询学习的自然语言语义查询技术实现知识图谱语义检索。

总结来说，知识图谱语义检索技术运用了知识图谱中测试工件的语义概念信息，结合自然语言处理技术实现知识语义检索，避免关键词索引带来的语义差异化问题，从而保证知识检索的准确性，为后续测试工件推送与相似测试工件检索奠定基础。

#### 4.1.1 基于关键词的知识图谱语义搜索技术

用户检索中的关键词可直观表达用户的信息需求，因此基于关键词的信息检索被广泛应用到各大商业搜索引擎，取得了很大的成功。因此，本项目首先研究基于关键词的知识图谱语义检索技术。

知识图谱上的关键词查询主要可以分为两类：基于关键词直接在知识图谱上搜索答案；基于关键词生成结构化的查询，进而提交给查询引擎得到结果。

##### (1) 语句关键词

为了获取用户查询语句中的关键词信息，本项目首先构建领域关键词词典，构建领域关键词库，然后训练领域关键词识别模型。针对用户查询语句，首先分词，获取初步关键词集合，进一步采用关键词识别模型识别关键词信息，并结合关键词拓展形成最终的关键词集合。

## (2) 基于关键词直接在知识图谱上搜索答案

在知识图谱上直接进行知识检索，其核心思想是采用知识图谱子图定位的策略。基本流程是在知识图谱上建立有效的关键词和知识图谱子图的索引，也就对实体、子图等进行索引。对于给定的关键字查询，首先在索引上匹配得到候选的知识图谱子图，实现对搜索空间的剪枝。最后，在小范围的知识图谱子图上进行搜索，找到最终的查询答案。该类方法的核心在于索引的构建，其构建方式直接决定搜索的效率和结果的质量。常见的索引方式有：

①实体关键词倒排索引。通过构建索引，快速定位知识图谱中包含关键词的实体。

②摘要索引。主要是构建一些包含结构化查询实体和关系类别的索引，在线上处理时根据类别摘要进行扩充。

③路径索引。主要借助关键词中包含的查询起始和终止结点，在图上按路径搜索提高查询效率。

基于关键词直接在知识图谱上搜索答案主要可以解决简单的知识语义搜索，即查询答案仅仅出现在单条知识图谱三元组中，对于复杂的语义查询往往无法适用。因此，本项目提出将关键词集合转化为结构化查询，从而解决上述问题。

## (2) 基于关键词生成结构化的查询

将关键词集合转化为结构化的查询方法主要包括三个步骤：

(1) 关键词映射。进行映射的主要原因是用户输入的关键词和知识图谱上的实体关系往往存在语义鸿沟，例如，关键词“问题缺陷”在知识

图谱可能对应的是“问题单”。所以，需要将关键词映射到知识图谱上实体、关系以及文本内容等。在此过程中，需要对知识图谱进行预处理，构建关键词和知识图谱实体和边的索引，进而在知识图谱上快速定位与关键词相关的实体和关系。

(2) 候选结构化查询构建。映射关键词后，生成了对应的实体和关系。在知识图谱中，基于生成的实体和关系拓展，能够生成局部的知识图谱子图，就得到了结构化查询需要的查询图结构。在此基础上，根据查询意图，将局部子图中的部分实体和关系替换为变量，进而生成结构化的查询。

(3) 候选结构化查询排序。在关键词映射过程中，一个关键词往往映射到知识图谱中的多个实体或关系，进而发现多个局部子图，生成多个结构化的查询。因此，需要对生成的结构化查询集合进行排序。例如，可以基于关键词搜索相似度、实体的拓扑度分布等指标来计算排序评分。

基于关键词的语义搜索还需要考虑对查询结果进行排序(如常见的TF/IDF方法)，让用户通过观察排序结果进而更新关键词。

#### 4.1.2 基于表示学习的知识图谱语义检索技术

近年来，知识图谱表示学习技术的出现，在知识图谱存储、构建、补全以及应用层面都产生了深远的影响。知识图谱表示学习旨在通过机器学习技术，将知识图谱中的实体和关系投射到连续低维的向量空间中，同时保持原有知识图谱的基本结构和性质。在知识图谱表示学习技术出现之前，通常以图数据库的形式组织和存储知识图谱。然而，随着开放知识图谱数据规模越来越大，即使是中等规模的知识图谱也可能包含了数以千计的关系类型、数百万的实体和数亿的三元组。传统的基于图存储和图算法的知识图谱应用越来越受限于数据稀疏性和计算效率低下的问题。

本项目研究基于表示学习的知识图谱语义检索技术，通过知识图谱表示学习技术，将其投射到低维连续的向量空间中，可有效提升语义检索性

能，主要包括基于表示学习的结构化语义查询和基于表示学习的自然语言语义查询。基于表示学习进行知识图谱检索有两大好处，一是在连续向量空间中，可以直接进行数值型计算，对查询术语或者关键字进行扩展，效率极高。例如，衡量两个实体之间的相似度可以通过直接计算两个实体在向量空间中的欧式距离来实现。二是低维连续的知识图谱向量表示是通过机器学习得到的，其学习过程既考虑了知识图谱的局部特征，又考虑了全局特征，生成的实体和关系的向量在本质上是一种蕴涵语义更丰富的表示，可以进行高效率的查询推理。

### (1) 基于表示学习的结构化语义查询

表示学习在结构化语义查询的应用主要是可以有效、高速地进行近似语义搜索。如下图所示，初始的结构化查询可以看作是一个查询图，虽然查询图中的查询目标在数据层中不存在，但可以基于查询图，利用翻译机制等表示学习算子计算出其在向量空间中的坐标（如下图中点 A 所示），进而通过最近邻搜索找到近似结果（如下图中点 B 所示），该近似结果很可能接近用户的初始查询意图。

查询图

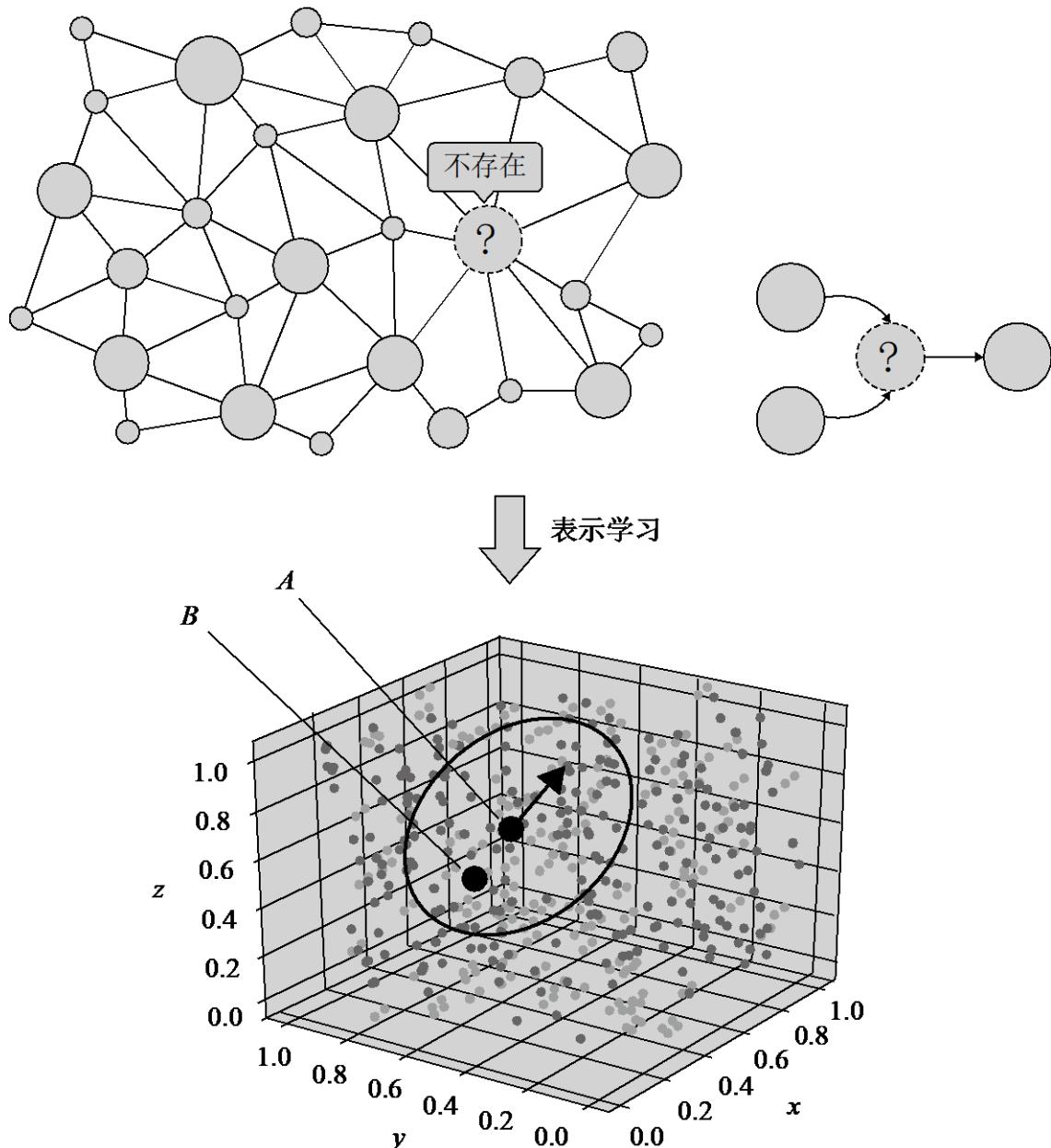


图 3-35 基于表示学习的知识图谱结构化查询示意图

## (2) 基于表示学习的自然语言语义查询

自然语言形式的语义查询的核心在于短语 (phrase) 到知识图谱上实体或边的映射，进而生成结构化的查询。在映射的过程中，主要难点在于关系 (实体之间的边或实体属性) 歧义的消除和查询图的构建。表示学习技术在这两个过程中都可以充分发挥作用。

首先，在离线阶段，生成知识图谱的实值向量，并且将关系短语词典和知识图谱中的关系在向量空间中对齐。在线上阶段，通过关键字检索的方式发现知识图谱中和自然语言短语对应的候选实体和边的集合。

知识表示学习将每一个候选实体集合中的实体平均实值向量作为查询图生成时的实体表示，进而并不需要某一个具体的实体向量。在计算查询图时，也可以利用翻译机制等原理提前预估查询图的评分好坏，提高生成效率和质量。

知识表示学习在传统语义检索的数据和查询之间提供了全新的向量空间维度，进而利用实值向量计算的优势对查询进行改进。以近似查询来说，基于表示学习的搜索可以在不修改初始查询的前提下直接返回近似结果，极大地提高近似查询的质量，为知识图谱近似查询提供了全新的思路。

## 4.2 基于知识图谱的测试工件智能推送技术

基于知识图谱的测试工件智能推送技术以知识图谱中存储的不同测试过程产生的测试工件为核心，制定不同测试过程下的推送内容，依据知识图谱分析出关键测试工件，如易出问题的测试项、测试用例，必需测试项等，计算知识图谱中节点重要程度，为后续测试工件推送奠定基础。基于知识图谱的测试工件智能推送技术包括测试工件重要度度量技术、面向测试过程的测试工件智能推送技术、基于知识图谱的测试用例可解释性推荐技术。

### 4.2.1 多策略测试工件重要度度量技术

在软件测试中，大部分软件测试用例不会产生问题单，只有少量的关键测试用例会产生问题单，同时不同测试项的重要程度不同，如何帮助测试人员进行引导性测试是本项目要解决的一大问题。要实现引导性测试，就需要依据软件测试知识图谱对测试工件进行重要度度量，从而可依据测试工件重要度在后续的测试工件推送中优先推送重要的测试工件，以达到引导性测试的目的。

软件测试知识图谱表征了测试工件、测试知识的关联，其中涵盖了“测试项->测试用例->问题单”路径信息，完整记录了发现问题的路径，而测试人员在进行测试设计和执行时需要更多考虑问题单，以便充分挖掘软件缺陷，提早发现软件质量问题。因此，本项目研究以问题单为导向的测试工件重要度度量技术，首先找寻知识图谱中所有“测试项->测试用例->问题单”，对发现问题的测试项进行命名实体识别，统计所有出问题的测试项的命名实体识别组合特征，分析获取出更易出问题的命名实体组合，形成命名实体组合模板，如“导弹型号”和“战绩指标”组合。

本项目为每一个命名实体组合模板分配权重，以表征每一模板对应的缺陷概率，权重越大，说明按照这一模板设计测试项、测试用例，有更大概率捕获软件异常。具体的权重分配和模板挖掘步骤如下所示：

步骤 1：获取知识图谱中“测试项->测试用例->问题单”等具有问题的路径链条；

步骤 2：对测试项、测试用例等进行命名实体识别，识别出测试项和测试用例等中的命名实体；

步骤 3：对测试项、测试用例中的命名实体类别（本体）组合进行统计分析，获取每一组合下出现的问题集合及每一个问题等级；

步骤 4：计算每一个命名实体本体组合对用的问题集合的权值，可按照出现不同问题的次数，问题等级等进行加权计算；

步骤 5：对命名实体类别组合进行权重排序，获取易出问题的命名实体类别组合，形成模板；

步骤 6：对软件测试知识图谱中的每一测试项、测试用例等，查看符合的模板，按照模板的权重计算获取每一测试项、测试用例的权重；对于其他测试工件，按照与它相连接的测试项和测试用例的权重，加权获取自身权重。

该算法从问题单的角度分析了每一测试工件的权重计算方法，结合了命名实体组合特征，体现了“以何种命名实体类别组合来设计测试项、测试用例才能发现问题”的思想，是一种典型的引导性测试。例如，测试人员发现如何测试项中同时包含“FPGA 型号名”和“战技指标”这一模板组合时，此时的测试项分级后的测试用例会有很大概率会问题，因此，如果可以挖掘出这些命名实体模板，那么按照这些命名实体模板设计测试用例可更大程度上挖掘出软件缺陷。

除了上述以问题单为导向的测试工件权重计算方法外，软件测试知识图谱可按照知识图谱中的 PageRank 算法来计算每一个实体的权重，从而依据图理论给测试工件进一步约束权重。

PageRank 算法是用于衡量图节点重要度的典型算法。在测试知识图谱中，如果与一个测试项相连的测试用例出现了更多的问题，则说明这一测试项是比较重要的测试项，因此本项目拟引入 PageRank 算法，对软件测试知识图谱中的节点权重进行计算。

#### 4.2.2 面向测试过程的测试工件智能推送技术

本项目开展面向测试过程的测试工件智能推送技术，首先依据不同类型项目、不同测试阶段所产生的测试工件特点，分析每阶段所需测试工件类型，确定评测过程各阶段需要推送的测试工件，实现测试工件智能化推送，在测试工件推送时，优先推送权重大的测试工件，辅助测试人员完成引导性测试。在测试工件推送时，不仅仅将测试工件节点推送给用户，而是将包含待推送测试工件的知识图谱局部子图推送给测试人员，这样可辅助测试人员理解推送内容，提高测试工件推送效果。

下面首先分析了测试工件，梳理了每一测试阶段重点关注的测试工件类型，最后说明了要推送的包含测试工件的知识图谱子图构建方法。

##### (1) 测试工件

本项目依据 GJB2725A-2001《测试实验室和校准实验室通用要求》等相关文件规定，结合第三方测试流程，将测试过程按照阶段划分为项目建立、测试需求分析、测试策划、测试设计与实现、测试执行、回归测试、测试总结、交付这八个部分。本项目在前述章节依据每个测试阶段特点、数据输入和输出，定义了不同阶段的测试工件，基于抽取的测试工件建立了软件测试知识图谱，将测试工件及相关关系有效表征，为后续测试工件推送奠定了基础。

### (2) 项目建立阶段

项目建立阶段是项目经理依据装备软件任务书、甲方测试需求等，填报项目的基本信息，形成测试项目信息这一测试工件。在此阶段，可从项目任务书、甲方需求中抽取项目基本属性信息，结合知识图谱中相似测试项目信息，自动抽取出项目基本信息，推送相似项目简述信息，辅助项目经理快速、正确完成项目建立，提高效率。项目建立后，形成新的测试项目信息节点，推送给相关测试人员，方便测试人员进行需求分析，形成测试需求规则说明。

### (3) 测试需求分析阶段

测试需求分析阶段主要是依据甲方需求文档、任务书、设计文档、项目基本信息等，分解测试需求，将测试需求分解为独立的测试项，形成测试需求规格说明。在这一阶段，测试需求分解是核心，推送重点是测试项推送，即推送出历史测试软件相似需求的测试项，以便测试人员依据测试需求，快速、准确完成测试项分解，在推送测试项时，将权重大的测试项及测试项对应的命名实体类别模板返回给测试人员，从而辅助测试人员依据历史测试项、命名实体模板来设计测试项。

软件测试知识图谱中存储着历史的历史测试工件，包括测试项目信息、软件需求节点信息、测试项信息，并且保存了需求节点与测试项节点的关联关系，这关系表征了不同测试需求向测试项的分解追溯情况。在测试人

员进行测试项分解，编写测试规则说明时，系统依据软件基本信息的相似性、需求相似性等，将软件知识图谱中的相似需求节点及对应的测试项节点等测试工件，采用关系表的形式推送给测试人员，从而帮助测试人员快速完成测试需求分析。

表 3-15 测试需求分析阶段推送信息表

测试项目信息	研制需求	测试项
项目信息节点 1	需求节点 1	测试项节点 1 测试项节点 2
	需求节点 2	测试项节点 3 测试项节点 4
	.....	.....
项目信息节点 2	需求节点 3	测试项节点 5 测试项节点 6
	需求节点 4	测试项节点 7 测试项节点 8
	.....	.....

#### (4) 测试策划

测试策划阶段根据软件测评任务书、合同或其他等效文件，以及软件需求规格说明和设计文档等进行计划，包括选择测试策略、确定测试方法、确定测试人员、分配项目组，制定任务分解信息（项目各阶段工作量）。测试策划阶段主要涉及的测试工件包括任务分解信息、人员画像、项目组、测试计划等。

任务分解信息包括预估的文档审查工作量、静态分析工作量、代码审查工作量、动态测试工作量，在进行这一部分时，系统依据前述形成的测试项目信息、测试项等测试工件，推送相似项目的实际任务完成信息，以辅助测试人员完成任务分解。

人员画像主要是对测试人员进行不同维度的能力评价,包括各领域测试能力、各软件类型测试能力、各编程语言测试能力、各阶段测试能力等,这些信息唯一表征着不同测试人员的业务水平,系统依据测试项目基本信息推送合适的测试人员,辅助完成精确的人员分配,保障项目顺利实施。推荐测试人员时,同时推送以往项目组信息,结合测试人员所分配的测试阶段,辅助完成项目组成员建立。

测试计划主要依据项目基本信息,形成测试策划、测试技术与方法、测试环境、测试进度、人员安排等,系统推送软件知识图谱中相似项目的测试计划节点,辅助测试人员快速完成指定测试计划。

### (5) 测试设计与实现

测试设计与实现主要是依据测试需求规格说明和测试计划,对测试项进行进一步分解,形成具体的测试用例。系统依据本项目测试项阶段,找寻知识图谱中相似测试项内容,结合项目基本信息,推送不同测试相对应的测试用例,辅助测试人员快速、准确完成测试项分解,形成测试用例。其次,本项目重点研究基于知识图谱的测试用例推荐技术,结合基于软件演化的测试用例生成技术,实现测试用例推送与生成。

在进行测试用例推送时,同时给出测试用例的来源信息,包括测试用例所属测试项、测试项目等,下表展示了测试用例推送表。

表 3-16 测试用例推送信息表

测试需求	测试项	测试用例	测试项目信息
需求节点 1	测试项节点 1	测试用例 1	所属测试项目
		测试用例 2	
	测试项节点 2	测试用例 3	
		测试用例 4	
	.....	.....	.....
需求节点 2	测试项节点 3	.....	.....
	测试项节点 4	.....	.....

.....	.....	.....	.....
-------	-------	-------	-------

### (6) 测试执行

测试执行阶段主要涉及到测试执行记录节点和测试问题单节点等，测试人员依据测试用例执行结果，形成测试执行记录单与问题单。系统软件知识图谱中存储的测试执行记录、测试问题单等，向测试人员推送历史测试问题单，辅助测试人员进行测试问题单形成。

### (7) 测试总结分析

测试总结分析阶段主要是依据测试过程数据，形成测试报告，依据前期形成的测试需求规则说明、测试计划、测试说明、测试项、测试用例、测试环境、问题单等信息，结合不同项目的具体模板格式，形成初步的测试报告，推送给测试人员，辅助测试人员完成测试总结分析工作。

总之，测试工件推送将依据测试工件自身特点、测试阶段特点进行推送，目的是辅助测试人员提高测试效率、提升测试质量，保障型号任务顺利进展。

### (8) 推送子图生成

单个测试工件不利于测试人员理解，缺乏必要的上下文信息。本项目在进行性测试工件推送时，首先生成待推送的测试工件集合，然后在知识图谱中找寻包含这些测试工件的知识图谱子图，作为最终要推送的知识图谱子图。例如，针对测试用例这一测试工件，本项目将与测试用例关联的测试项、测试需求、软件信息等关联节点纳入到子图中。

#### 4.2.3 基于知识图谱的测试用例可解释性推荐技术

测试用例是动态测试的输入，测试人员需要依据测试项进行测试用例分解。军用装备软件的相似需求对应的测试用例具有一定的重叠性，因此可依据软件来辅助测试人员生成测试用例。具体而言，采用深度学习相似度模型计算测试需求间语义相似度，在知识图谱中查询与用户提供的测试需求语义相关的测试需求节点，找寻与用户需求相关的测试用例，并筛选

易出问题的测试用例，为测试人员推荐更易发现缺陷的测试用例，提高软件测试的缺陷发现率，保证软件测试的质量。本项目拟突破基于知识图谱的测试用例可解释性推荐技术，提高测试自动化程度，提升测试知识复用度，具体方案在技术途径处详细说明。

### 4.3 基于模板的测试用例自动生成技术

测试用例生成是测试阶段重要的环节，需要花费大量的人工成本去做测试用例的生成。测试用例复用是目前解决测试用例生成的一大技术手段，通常相似功能的系统在部分业务场景上具有相似的数据流转特点，因此可将历史测试用例进行修正和扩充，以生成面向新测试需求的测试用例，可大大降低测试用例生成的成本，并且可对测试用例进行优先等级排序，以提高测试质量和效率。

本项目研究基于模板的测试用例自动生成技术，旨在将以往测试知识充分利用，挖掘测试项关键信息，结合测试项模板和测试用例模板生成测试用例。基于模板的测试用例生成技术包括测试用例模板挖掘技术和基于关键信息的测试用例生成技术。首先，测试用例模板挖掘技术对测试项和测试用例进行分析，挖掘测试项和测试用例中的关键命名实体信息，如软件名称、操作名称、周期时间等，分析出测试项和测试用例中的关键槽位置，形成测试项-测试用例模板。对于新的测试项内容，依据模板挖掘获取测试项中的关键参数信息，进而填充到测试用例模板以生成测试用例。

#### 4.3.1 测试用例模板挖掘技术

测试人员依据测试项描述信息，识别测试项中的关键信息，进而分解测试项得到测试用例。测试项在描述时，具有一定的模板性，如“中断 6：…做 ZD 下位机 ZZ 软件内部时序，周期为 30ms”，其中“中断 6”、“周期”和“30ms”是设计测试用例时的关键信息，可将其抽取为模板形式“<slot\_中断>\*周期<slot\_时间>\*”，<slot\_中断>和<slot\_时间>即槽位置信息，表示在相似测试项中可以变化的部分，这个模板表示对中断的周期描述的

测试项模板。在后续的需求演化后，如果出现了上述中断周期描述模板，则首选获取新测试项中的<slot\_中断>和<slot\_时间>，从而得到了中断名称和中断周期时间这两个关键参数信息，将中断名称和中断周期时间填入到对应的测试用例模板描述中，从而辅助测试人员获取初步的测试用例。

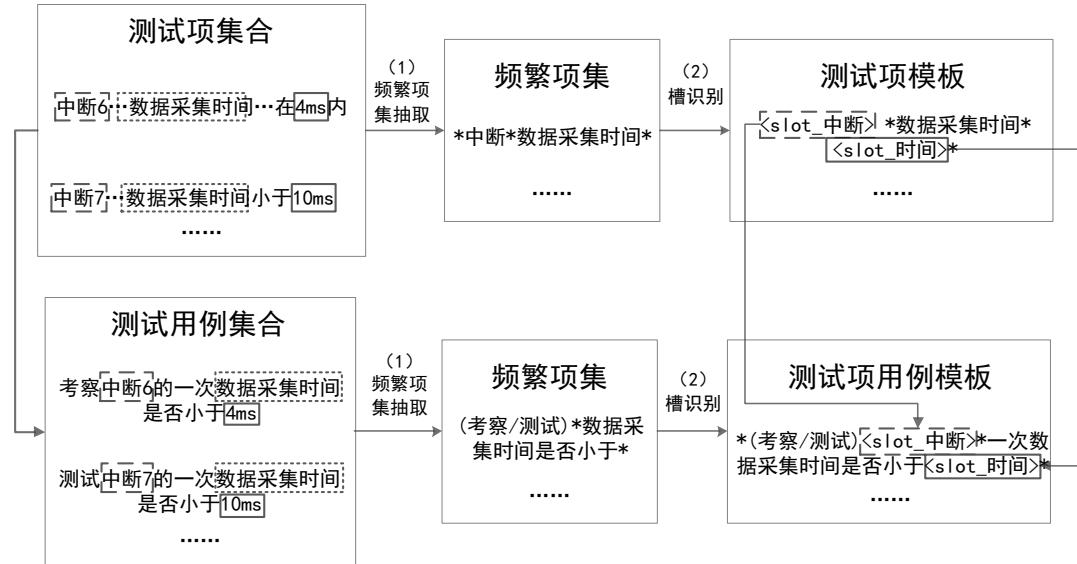


图 3-36 模板挖掘示意图

本项目拟采用基于 Apriori 的模板挖掘算法，从文本中挖掘获取模板，具体方案图如下所示：

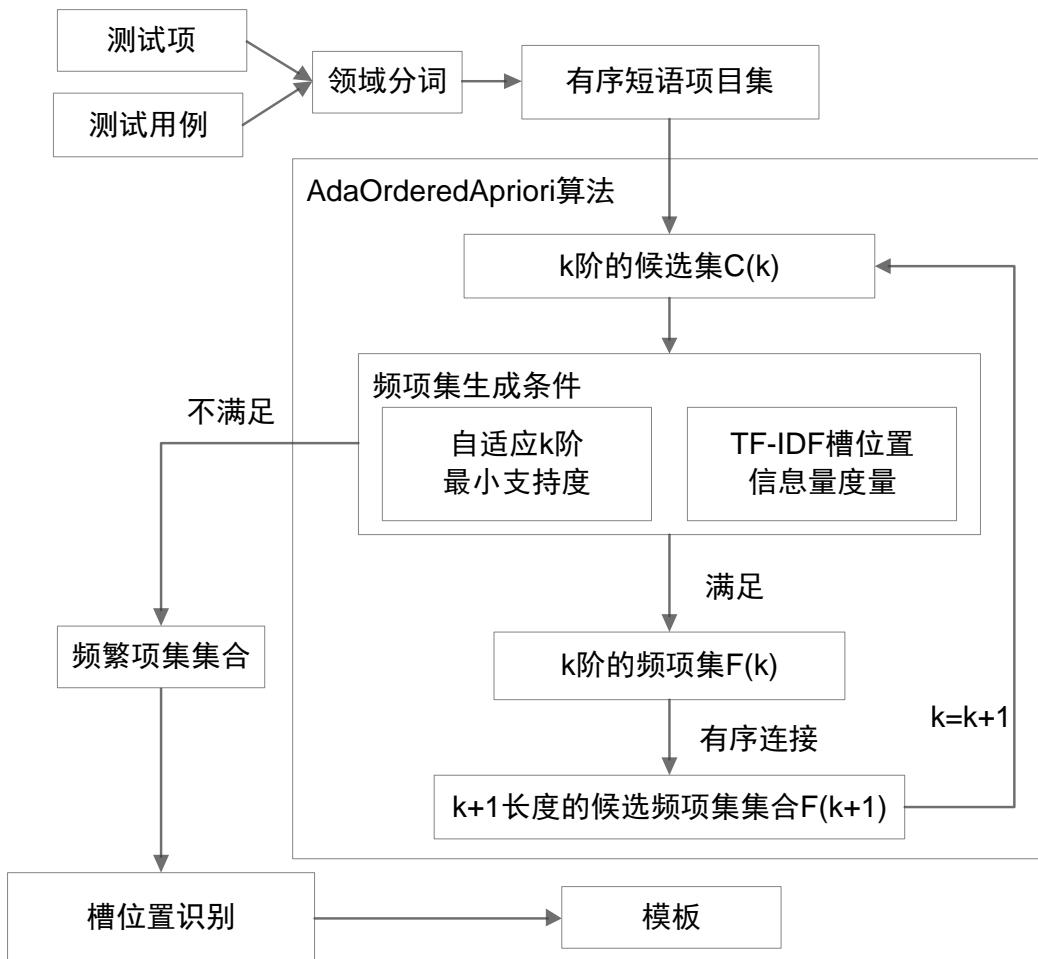


图 3-37 模板挖掘算法示意图

本项目引入的算法如上图所示，通过迭代生成和过滤获得所有满足条件的频繁项，第一步，基于领域词库对测试项、测试用例进行分词，作为算法的输入；第二步，对短语项进行有序连接，获取优选的候选项集；第三步，采用自适应机制，随着频繁项集的项数增加，支持度自适应下降，以获得较长的频繁项集；第四步，使用 TF-IDF 表征候选项集中间隔词的混乱程度，剔除低信息量频繁项集；最后，将算法输出的频繁项作为模板的框架词，可用于槽识别获取模板。

本项目引入上述算法实现了测试项、测试用例模板挖掘，并将测试项模板与测试用例模板对应起来，为后续依据测试项关键信息生成测试用例提供支撑。

#### 4.3.2 基于关键信息的测试用例生成技术

本项目在测试项模板、测试用例模板基础上，实现基于关键信息的测试用例生成技术。对于先来的测试项，首先识别其中的命名实体信息，将命名实体转化为对应的本体标签，进一步判断测试项所符合的测试项模板，依据测试项模板挖掘获取关键信息，然后依据关键信息和测试用例模板实现测试用例生成，即将关键信息填充到具体的测试用例中，以生成初步的测试用例。

如下图所示，对于一个新的测试项“中断 8：…要求数据采集时间不大于 12ms…”，则首先从这一测试中抽取获取“中断 8”、“数据采集时间”“12ms”关键信息，进一步将“中断 8”替换为槽位置<slot\_中断>，12ms 替换为<slot\_时间>，找到匹配的测试项模板<slot\_中断>\*数据采集时间\*<slot\_时间>\*。然后，根据测试项模板对应的测试用例模板，将关键信息“中断 8”和“12ms”填充到对应的位置，生成最终的测试用例“测试中断 8 的依次数据采集是否小于 12ms”。

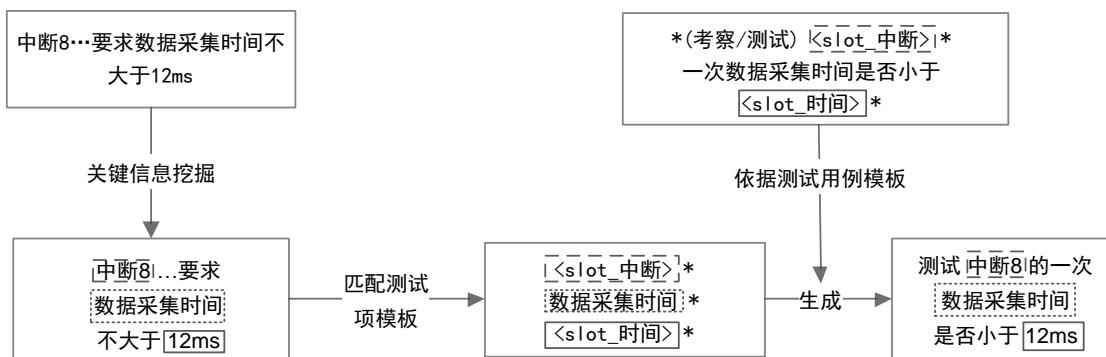


图 3-38 测试用例生成示例图

#### 5. 典型装备软件质量评价技术

典型装备软件质量评价技术总体方案如下图所示，总体方案体系结构由装备软件质量度量元分析技术、装备软件质量度量元归一化技术和装备软件质量综合度量与评估技术组成。

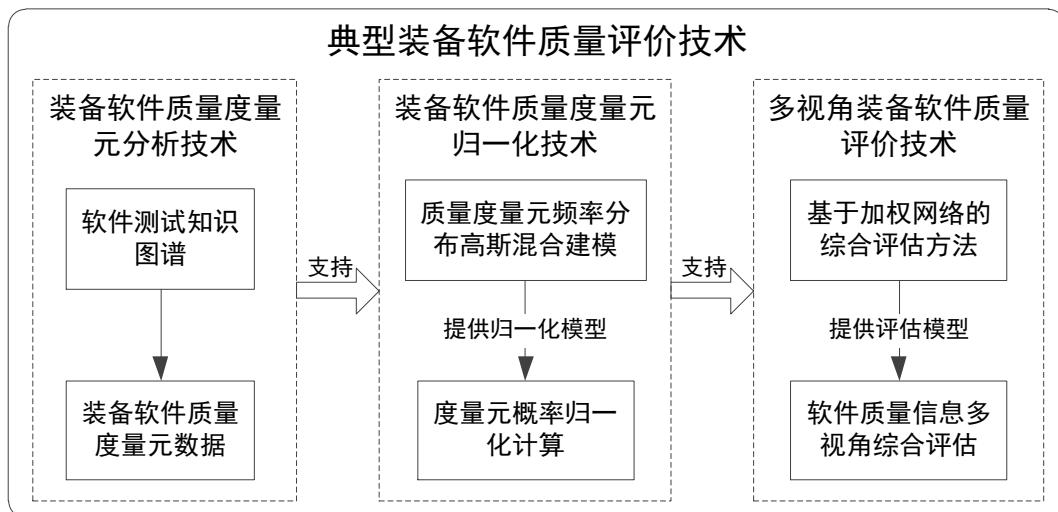


图 3-39 典型装备软件质量评价技术总体方案

首先，以通用软件质量模型为基础，定义装备软件质量度量元，进一步，分析软件测试知识图谱中与度量元相关的数据，从知识图谱中抽取装备软件质量度量元数据信息，为软件质量度量与评估提供数据支撑。

其次，开展装备软件质量度量元归一化技术研究。应用高斯混合模型对度量元进行概率建模，得到不同质量度量元的概率分布，并对度量元进行归一化计算，消除不同质量度量元之间的量纲影响，从而为后续装备软件质量评估提供特征。

最后，开展多视角装备软件质量评价技术研究，根据不同质量度量元间的关系，构建质量度量元相关关系网络，识别影响软件质量的关键少数质量度量元，根据不同的质量度量元权重分配进行软件质量的整体评估，并能够从时间、院所、型号、领域、阶段等视角综合评估软件质量。

## 5.1 装备软件质量度量元分析技术

装备软件质量度量元分析技术是装备软件质量度量的基础。装备软件质量与装备软件开发过程数据、第三方评测工程数据等有紧密的关联性，基于这些数据提取度量元信息，从而对装备软件质量进行评估，可有效指导装备软件迭代开发，提升装备软件质量。

### 5.1.1 装备软件质量度量元定义

装备软件的过程文档、软件代码、第三方评测过程数据作为装备软件研发和评测过程中的产物，其中存储了大量的有效软件质量信息，这些质量信息能充分反映装备软件质量。

本项目首先分析可表征装备软件质量的可量化指标，并将这些指标作为软件质量度量元。通用的软件质量度量评价标准在专业研究方面取得了不少成果，但却不能很好地应用到工程实践中，存在的主要问题有：1) 通用的软件质量度量模型追求的是大而全，覆盖面广，使得这些质量模型缺乏针对性，如《GJB 5236 军用软件质量度量》的外部度量和内部度量质量模型缺少了安全性这个军用软件的重要质量特性；2) 现有的软件质量度量没有对软件的开发过程和管理提供必要的反馈信息，质量度量的结果只是用于做事后评价，并没有达到提高软件质量的目的。本项目首先树立了《GJB 5236 军用软件质量度量》质量模型，如下图所示。

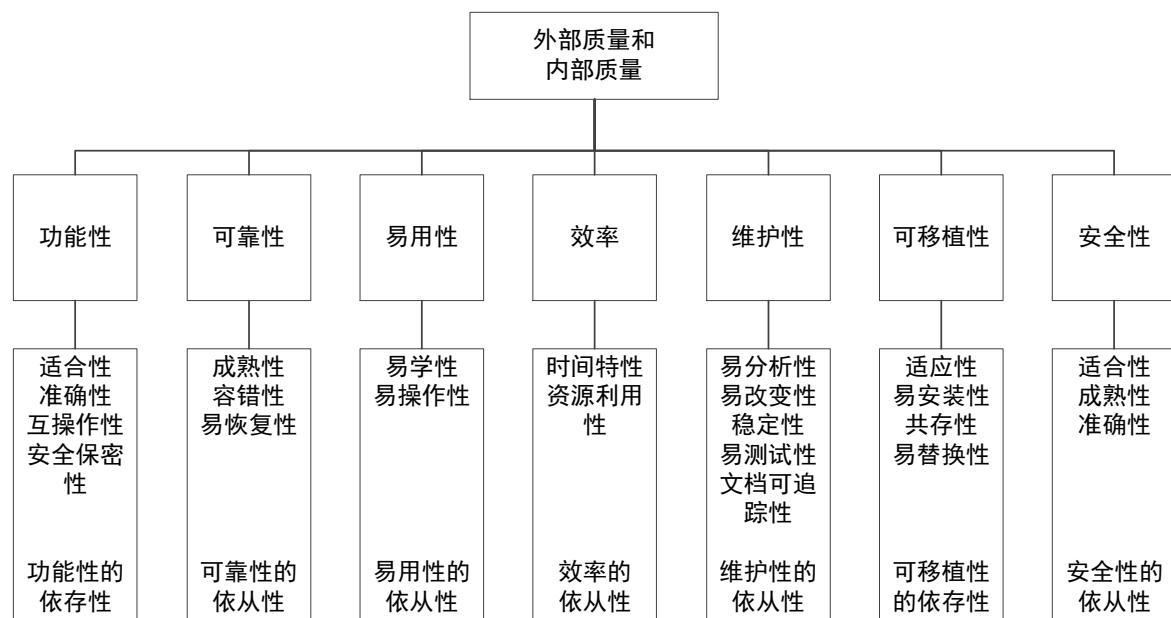


图 3-40 丰富后的质量模型

根据以上对质量模型，结合装备软件第三方测试过程数据，初步建立装备软件的质量度量元，结果如下表。

表 3-17 装备软件质量度量元

量化测试方法	质量度量元
需求和设计文档审查	软件需求的稳定性
	单个需求的变更数最大值度量
	千行代码文档问题数
	文档问题数
静态分析	模块的平均圈复杂度
	源文件个数
	头文件个数
	总行数(包括空行)
	总注释行数
	总注释率
	文件注释率小于 20% 的比例
	文件注释率
	模块数
	模块的平均圈复杂度
	模块的最大圈复杂度
	模块的圈复杂度比例控制度量
	模块平均规模行
	模块最大规模行
	模块平均传参数个数
动态测试	软件模块最大扇入扇出数
	模块规模(大于 200 行)过大的比例
	千行代码静态问题数
	静态问题总数
	千行代码程序问题数
动态测试	千行代码关键/重要程序问题数
	程序问题密度度量

	问题一次解决率度量
--	-----------

基于装备软件多源质量数据，定义装备软件度量元，涵盖了丰富的质量模型中质量特性、质量子特性的度量指标，为软件质量分析与评价提供可验证、可量化的度量依据。

### 5.1.2 装备软件质量度量元数据获取

软件测试知识图谱涵盖了软件开发文档、软件第三方测试过程数据，包含了需求和设计文档审查、静态分析、动态测试等多方面的知识，具有“软件→需求→测试项→测试用例→问题单”知识链条数据，可非常方便抽取各个质量度量元数据。

针对质量度量元，本项目结合软件测试知识图谱，给出如下的质量数据抽取策略：

表 3-18 质量度量元数据抽取策略表

量化测试方法	质量度量元	数据抽取策略（对应某软件）
需求和设计文档审查	软件需求的稳定性	(知识图谱既有具有更改标记的需求模块) / (知识图谱中所有需求模块)
	单个需求的变更数最大值度量	知识图谱中软件需求节点最大更改次数
	千行代码文档问题数	(知识图谱中文档问题) / (知识图谱中软件代码行数/1000)
	文档问题数	知识图谱中文档问题
静态分析	源文件个数	知识图谱中源文件个数
	头文件个数	知识图谱中头文件个数
	总行数(包括空行)	知识图谱中总行数(包括空行)
	总注释行数	知识图谱中总注释行数
	总注释率	(知识图谱中总注释行数) / (知识图谱中总行数(包括空行))
	文件注释率小于 20% 的比例	(知识图谱中注释率小于 20% 的文件) /

		(知识图谱中项目所有文件)
	软件模块数	知识图谱中软件模块数
	模块的平均圈复杂度	(知识图谱中所有模块圈复杂度的和) / (知识图谱中所有模块个数)
	模块的最大圈复杂度	知识图谱中所有模块的最大圈复杂度
	模块平均规模行	(知识图谱中软件代码总行数) / (知识图谱中所有模块个数)
	模块最大规模行	知识图谱中所有模块的最大规模行
	模块平均传参个数	(知识图谱中所有模块参数个数和) / (知识图谱中所有模块个数)
	模块最大扇入扇出数	知识图谱中的模块的最大扇入扇出数
	模块规模(大于 200 行)过大的比例	(知识图谱中代码行数大于 200 行的模块数) / (知识图中所有模块个数)
	千行代码静态问题数	(知识图谱中静态问题总数) / (知识图谱中软件代码行数/1000)
	静态问题数	知识图谱中静态问题总数
动态测试	千行代码程序问题数	(知识图谱中程序问题数) / (知识图谱中软件代码行数/1000)
	千行代码关键/重要程序问题数	(知识图谱中关键/重要程序问题数) / (知识图谱中软件代码行数/1000)
	程序问题密度	(知识图谱中程序问题总数) / (知识图谱中模块总数)
	问题解决率	(知识图谱中程序问题数-知识图谱中遗留问题个数)/ 知识图谱中程序问题数)

如下图所示为从软件测试知识图谱中抽取软件质量度量元的示例：

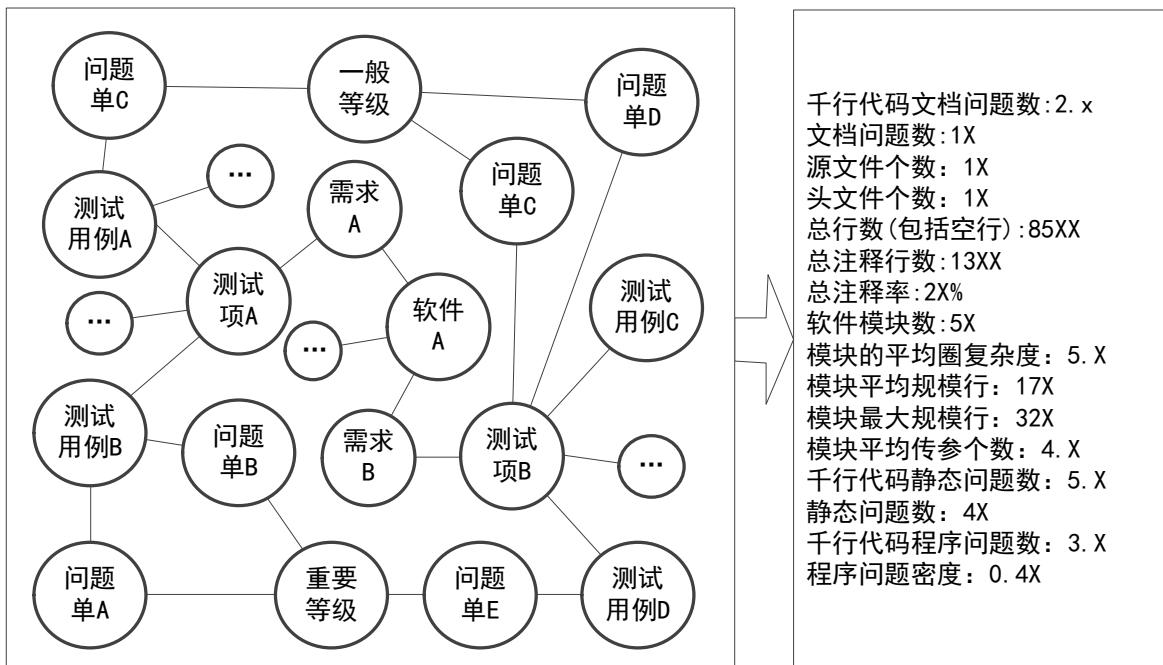


图 3-41 装备软件质量度量元数据抽取示例图

## 5.2 装备软件质量度量元归一化技术

不同装备软件质量度量元数据往往具有不同的量纲, 难以直接用于软件质量的综合评估。为实现装备软件质量的综合评估, 需要对度量数据进行标准化处理, 消除不同质量度量元之间的量纲影响, 统一各指标处于同一数量级, 为关键质量度量元发现与软件质量多视角综合评估提供数据基础。

为实现软件质量度量元的归一化, 本项目使用一种基于高斯混合模型的质量度量元概率归一化算法, 从概率的角度分别对数据中的各个质量度量元进行概率归一化处理。首先, 针对质量信息中的各质量度量元的数据分布, 使用无监督的 EM 算法对其进行高斯混合建模, 拟合该质量度量元的概率密度函数, 然后, 基于该质量度量元的概率密度函数计算其累积分布函数, 将质量度量数据归一化至  $[0, 1]$  或  $(0, 1)$  范围内。然后对各质量度量元进行分析, 若存在度量数据越小质量越好的质量度量元  $q_j$ , 则令  $q_j = 1 - q_j$ , 由此即实现了任意软件度量样本的归一化, 且度量属性归一

化后的值越趋近于 1 表示该度量属性质量水平越高。归一化后的软件质量数据能够支持软件质量的综合评估。

### 5.2.1 质量度量元频率分布高斯混合建模

对软件质量度量元进行归一化，首先需要对软件质量数据进行建模，求得最佳模型表征软件指数数据的特点。高斯混合模型是多个高斯分布函数的线性组合，理论上 GMM 可以拟合任意类型的分布。因此，本项目采用高斯混合模型进行质量数据的建模。

对质量度量元的度量值的分布进行高斯混合建模时，选择样本数据相对熵最小时超参数  $k$  值作为最终高斯混合模型的  $k$  值。选择算法如下：针对各个  $k$  值，首先，使用无监督的 EM 算法进行质量度量元度量值分布的高斯混合建模，然后，计算样本数据与所建模型的相对熵，相对熵最小高斯混合模型作为所求高斯混合模型。

设软件质量样本集  $X = \{X_1, X_2, \dots, X_n\}$ ，其中  $X_i = \{m_1^i, m_2^i, \dots, m_p^i\}$  是第  $i$  个软件质量度量样本，每个软件质量度量样本包括  $p$  个质量度量元； $m_j^i$  记为第  $i$  个软件样本的第  $j$  个质量度量元的度量结果。以第  $j$  个质量度量元为例介绍高斯混合模型的构建：

步骤 1：首先构造样本数据的第  $j$  个质量度量元数据集  $S_j$ ：

$$S_j = \{m_j^1, m_j^2, \dots, m_j^n\}$$

其中， $m_j^i$  表示第  $i$  个样本中第  $j$  维度量元的度量值；

步骤 2：初始化高斯混合模型的分量个数为  $k_j$ ，建立高斯混合模型  $GMM_{k_j}(\pi_j, \mu_j, \Sigma_j)$ ，则有：

$$GMM_{k_j}(\pi_j, \mu_j, \Sigma_j) = \sum_{k=1}^{k_j} \pi_k N(x | \mu_k, \Sigma_k)$$

步骤 3：计算高斯混合模型  $GMM_{k_j}(\pi_j, \mu_j, \Sigma_j)$  与真实样本的相对熵  $D_{k_j}$ ；

步骤 4：从  $k_j=1$  开始不断增大  $k_j$  值，并重复步骤（2）、（3），即可得到不同  $k_j$  值下的相对熵  $D_{k_j}$ ，根据  $D_{k_j}$  跟随  $k_j$  变化趋势，求出最佳的  $k_j$  值，

设为  $K_j$ 。则质量度量元  $j$  的高斯混合模型  
 $GMM_j(\pi_j, \mu_j, \Sigma_j) = GMM_{K_j}(\pi_j, \mu_j, \Sigma_j)$ ;

步骤 5：依次构建各质量度量元数据集，重复步骤 (1) ~ (4)，得到每个度量元  $j$  的高斯混合模型  $GMM_j$ 。

由此，即完成了质量度量元的高斯混合模型构建。

### 5.2.2 度量元概率归一化计算

基于质量度量元频率分布高斯混合建模构建的各质量度量元的高斯混合模型本质上是度量元的概率密度分布模型，因而基于概率统计分布函数相关概念，可以方便的实现质量度量元的范围归一化计算，将质量度量数据归一化至  $[0, 1]$  或  $(0, 1]$  范围内。

分布函数定义为随机变量  $X$  取值小于  $x$  的累积概率，也称为概率累积函数。使用高斯混合分布拟合特征  $X$  的概率密度函数  $p(x)$  时， $X$  的分布函数为：

$$F(x) = \int_{-\infty}^x p(x) dx$$

分布函数是随机变量最重要的概率特征，可以完整地描述随机变量的统计规律，并且决定随机变量的一切其他概率特征，此外，分布函数具有非降性、有界性和右连续性，其取值范围是  $[0, 1]$ ，具有良好的统计特性，故使用特征  $x$  的分布函数值  $F(x)$  作为特征  $X$  的归一化处理后的值，实现特征  $X$  的概率归一化计算， $X$  取  $x$  值的概率越大时，可以认为  $x$  值越重要。

类似的，在软件质量评估中，通过质量度量元的概率归一化计算，可将软件质量属性的度量值用于表征软件的质量水平。利用软件度量元的高斯混合模型，对于样本数据  $x = \{m_1, m_2, \dots, m_p\}$ ，其中  $m_j$  表示度量元  $j$  的质量属性值，计算其在对应高斯混合模型  $GMM_j(\pi_j, \mu_j, \Sigma_j)$  中的累积概率，从而得到该样本在度量元  $j$  的软件质量水平  $q_j$ ：

$$q_j = \int_{-\infty}^{m_j} \sum_{k=1}^{K_j} \pi_k \frac{1}{\sqrt{2\pi}\mu_k} \exp\left(-\frac{(x-\mu_k)^2}{2\Sigma_k^2}\right) dm$$

由此，即可得到任意软件度量样本 $x=\{m_1, m_2, \dots, m_p\}$ 基于高斯混合模型的度量元概率归一化结果：

$$q = \{q_1, q_2, \dots, q_p\}$$

从而实现了质量度量数据的范围归一化。

### 5.3 多视角装备软件质量评价技术

本项目基于质量度量元归一化结果，首先计算不同度量元之间的相关性，并计算不同度量元的权重，以此构建质量度量元加权关系网络，提出基于加权网络的软件质量综合评估方法；最后，根据软件质量评估层次、角度的多样性提出了软件质量多视角综合评估方法。

#### 5.3.1 装备软件关键质量度量元选择

软件质量度量元归一化度量实现了软件质量数据数量级、量纲的统一，能够实现对任意软件单一质量度量元的评估。然而，软件整体质量的评估需要综合各软件质量度量元的影响，找出影响软件整体质量的关键质量度量元，基于不同质量度量元的重要程度为不同软件质量度量元分配不同权重以实现软件复杂度综合评估。为评估不同质量度量元对于装备软件整体质量的重要程度，提出装备软件关键质量度量元选择技术，研究质量度量元之间涌现特征，发掘各质量度量元的相关关系，并构建软件质量度量元相关关系网络，基于复杂网络相关理论进行软件质量关键度量元的选择，是软件质量综合评估的关键过程，为软件的质量综合评估提供支撑。软件关键质量度量元选择技术的实现流程如下图所示。

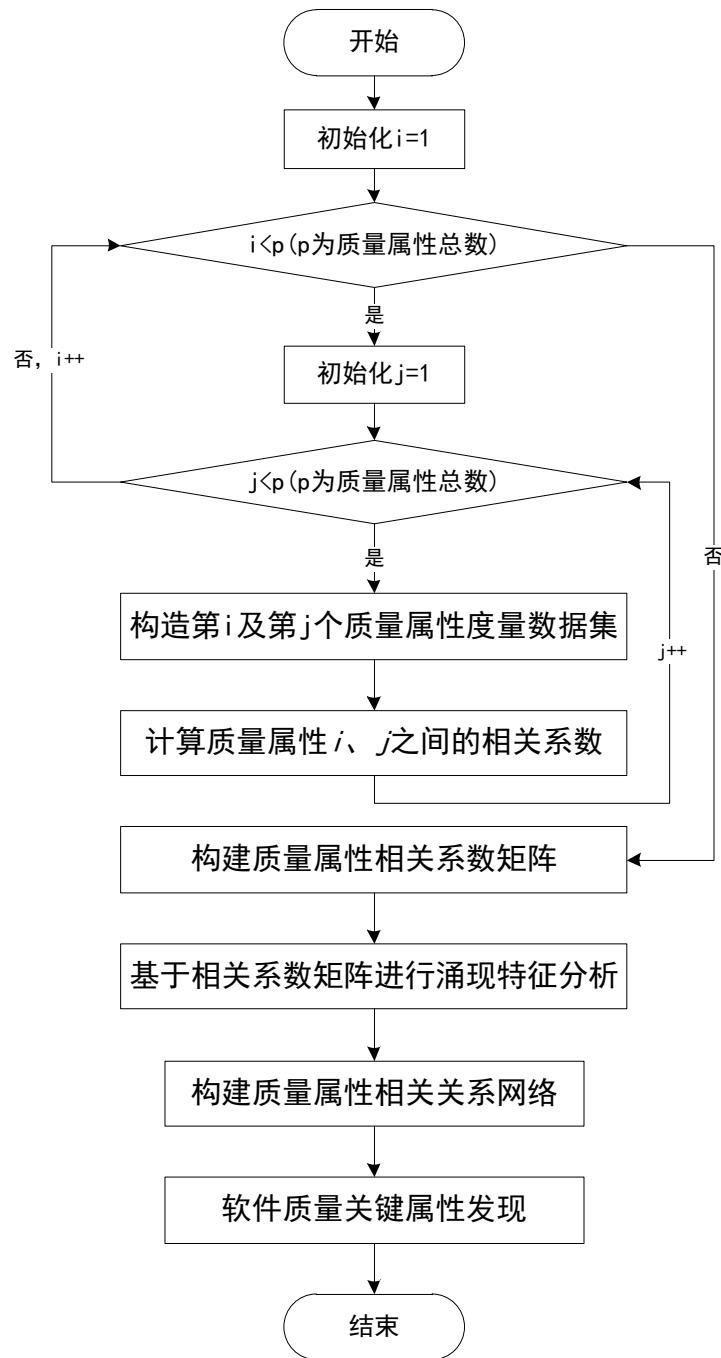


图 3-42 软件关键质量度量元选择技术实现流程图

### (1) 软件质量度量元涌现特征分析

软件质量模型中,某些质量度量元的变化可能会引起其它度量元的变化,研究质量度量元的涌现特征,即是研究不同软件质量度量元间的相互影响。

不同软件质量度量元间的相互影响主要分为三种类型：正相关、负相关，以及可能存在的某种函数关系。为简化研究过程，本项目中只当两质量度量元正相关，且两度量元间存在线性关系时，本项目认为两个度量元存在相关关系，其余情况均认为两质量度量元无相关关系。

为研究质量度量元的涌现特征，使用归一化之后的软件质量数据，具体过程如下：

步骤 1：构造第  $i$  维及第  $j$  维质量度量元度量数据集  $C_i$  和  $C_j$ ：

$$C_i = \{h_i^1, h_i^2, \dots, h_i^n\}, C_j = \{h_j^1, h_j^2, \dots, h_j^n\}$$

其中  $i, j \in [1, p]$ , 且  $i, j \in N^+$ ;

步骤 2：使用相关系数公式  $\rho_{ij} = Cov(C_i, C_j) / \sqrt{D(C_i)D(C_j)}$  计算质量度量元  $i$ 、 $j$  之间的相关系数，得到第  $i$  及第  $j$  个质量度量元之间的相关系数数值为  $\rho_{ij}$ ；

步骤 3：遍历  $i$ 、 $j$  值，重复步骤 (1) ~ (2)，得到质量度量元  $i$ 、 $j$  之间的相关系数矩阵  $\rho$ ：

$$\rho = \begin{bmatrix} \rho_{11}, \rho_{12}, \dots, \rho_{1p} \\ \rho_{21}, \rho_{22}, \dots, \rho_{2p} \\ \vdots \\ \rho_{p1}, \rho_{p2}, \dots, \rho_{pp} \end{bmatrix}$$

其中  $\rho_{ij}$  表示  $i$ 、 $j$  两个度量元之间的相关系数；

步骤 4：遍历矩阵  $\rho$  中的  $\rho_{ij}$  值，若  $\rho_{ij}$  小于设定的阈值  $\beta_1$ ，则将矩阵中的  $\rho_{ij}$  用 0 替代，得到新矩阵  $\rho'$ ：

$$\rho' = \begin{bmatrix} \rho_{11}', \rho_{12}', \dots, \rho_{1p}' \\ \rho_{21}', \rho_{22}', \dots, \rho_{2p}' \\ \vdots \\ \rho_{p1}', \rho_{p2}', \dots, \rho_{pp}' \end{bmatrix}$$

其中， $\rho_{ij}' = \begin{cases} 0, & \rho_{ij} < \beta_1, \\ \rho_{ij}, & \rho_{ij} \geq \beta_1; \end{cases}$

步骤 5：遍历步骤 4 中得到的新矩阵  $\rho'$ ，若  $\rho_{ij}'$  为正值，则求  $\rho_{ij}'$  对应的度量元  $i, j$  之间的斜率，得到斜率矩阵  $T$ ：

$$T = \begin{bmatrix} t_{11}, & t_{12}, & \dots, & t_{1p} \\ t_{21}, & t_{22}, & \dots, & \rho_{2p} \\ \vdots & & & \vdots \\ t_{p1}, & t_{p2}, & \dots, & t_{pp} \end{bmatrix}$$

其中  $t_{ij}$  表示  $i, j$  两个度量元之间的斜率；

步骤 6：遍历步骤 5 中得到的矩阵  $T$ ，若  $t_{ij} < 1$ ，则令  $t_{ij} = 0$ ，得到新矩阵  $T'$ ：

$$T' = \begin{bmatrix} t_{11}', & t_{12}', & \dots, & t_{1p}' \\ t_{21}', & t_{22}', & \dots, & t_{2p}' \\ \vdots & & & \vdots \\ t_{p1}', & t_{p2}', & \dots, & t_{pp}' \end{bmatrix}$$

$$\text{其中, } t_{ij}' = \begin{cases} 0 & t_{ij} < 1 \\ t_{ij} & \geq 1 \end{cases}$$

由此，即实现了软件质量度量元涌现特征的分析。对任意质量度量元  $i, j$ ，若  $t_{ij}' \geq 1$ ，则称质量度量元  $i, j$  相关，相关程度为  $t_{ij}'$ 。

## (2) 软件质量关键度量元发现

软件质量度量元涌现特征分析研究了不同软件质量度量元间的相互影响，基于此可以实现软件质量关键度量元的发现。软件质量度量元涌现特征分析得到了质量度量元相关关系矩阵  $T'$ ，以  $T'$  作为邻接矩阵构造软件质量度量元相关关系网络  $G_0$ 。软件质量度量元相关关系网络  $G_0$  为一个有向有环图，节点代表各个质量度量元节点，边表征质量度量元的相关关系，权重等于相关程度值  $t$ 。为找出软件质量关键度量元，首先基于经典的深度优先搜索算法找到  $G_0$  中的环，去除环中使得整体权重下降最小的边，将有向有环图  $G_0$  转化为有向无环图  $G$ 。本项目定义软件质量关键度量元

为有向无环图  $G$  中入度为 0 的质量度量元节点，如下图所示，蓝色显示的节点即为本项目中的关键质量度量元节点。

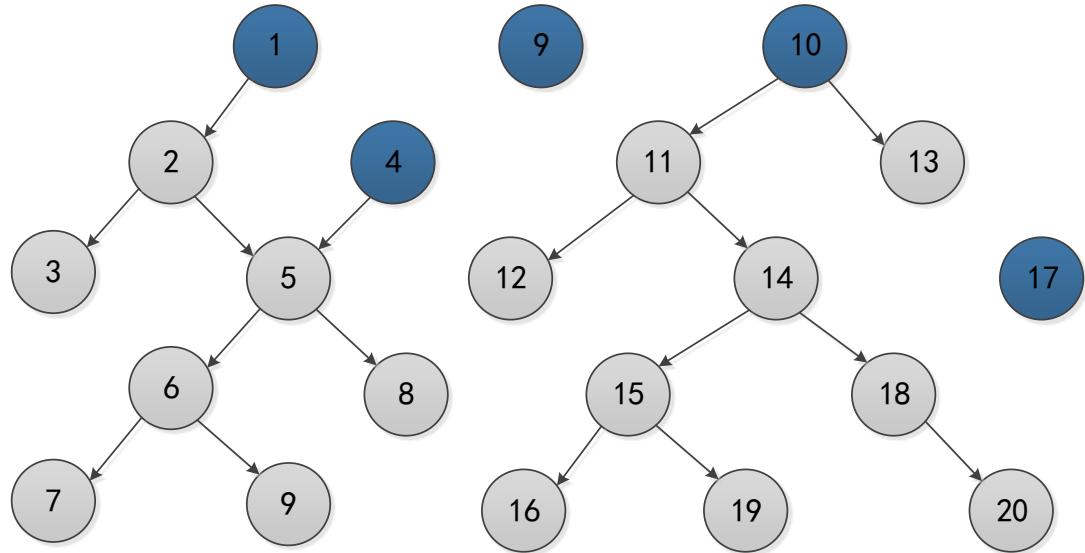


图 3-43 关键质量度量元确定方法

由此，本项目得到了修正的软件质量度量元相关关系网络  $G$ ，并找出影响软件质量的关键少数质量度量元。

### 5.3.2 基于加权网络的综合评估方法

本节以构建的质量度量元相关关系复杂网络为基础，提出一种基于加权网络的软件质量综合评估方法。基于加权网络相关方法计算出各质量度量元的权重，而后通过加权求和法计算软件整体质量水平。质量度量元相关关系复杂网络可以视为一种以相关系数为权重的加权网络。在加权网络中，连边的权重被视为节点间交互作用的强度。节点  $i$  的强度  $s_i$  定义为：

$$s_i = \sum_{j \in N_i} w_{ij}$$

其中  $N_i$  为  $i$  临近节点的集合， $w_{ij}$  表示  $i$ 、 $j$  连边的权重，也即质量度量元  $i$ 、 $j$  的相关系数。为了实现软件质量的综合评估，需要基于软件质量度量元对软件质量的重要程度分配不同的权重。本项目定义质量度量元  $i$  的影响域  $I_i$  表征软件质量度量元对于软件质量的重要程度。当质量度量元  $i$  在加权网络中不是孤立点时，

$$I_i = \begin{cases} 1, & \text{当质量度量元 } i \text{ 无后续节点时;} \\ \sum w_{ij} \times I_j, & \text{当质量度量元 } i \text{ 有后续节点时。} \end{cases}$$

当质量度量元  $i$  在加权网络中为孤立点时，其影响域：

$$I_i = \frac{\sum_{k \in N_c} I_k}{\text{num}(N_c)}$$

其中  $N_c$  为非孤立点集合， $\text{num}(N_c)$  表示  $N_c$  中元素的个数。由此可计算各质量度量元的权重：

$$W_i = \frac{I_i}{\sum I_i}$$

其中， $W_i$  表示质量度量元  $i$  的权重。

由此可以得到，软件质量的综合评估结果：

$$R = \sum W_i \times R_i$$

其中  $R_i$  表示质量度量元  $i$  的归一化的度量结果。

### 5.3.3 软件质量信息多视角综合评估

软件质量评估不仅要基于质量信息评估软件当前的质量状况，更要从多个视角对软件质量进行评估。随着评估角度、评估层次的不同，装备软件质量信息的评估结果存在巨大差异。因此软件质量的评估需要基于评估角度、评估目的展开。本项目提出软件质量信息多视角综合评估方法，选用质量评估中几个最重要的视角对软件质量信息进行评估。主要包括时间、型号、领域、院所及混合视角，如下图所示。

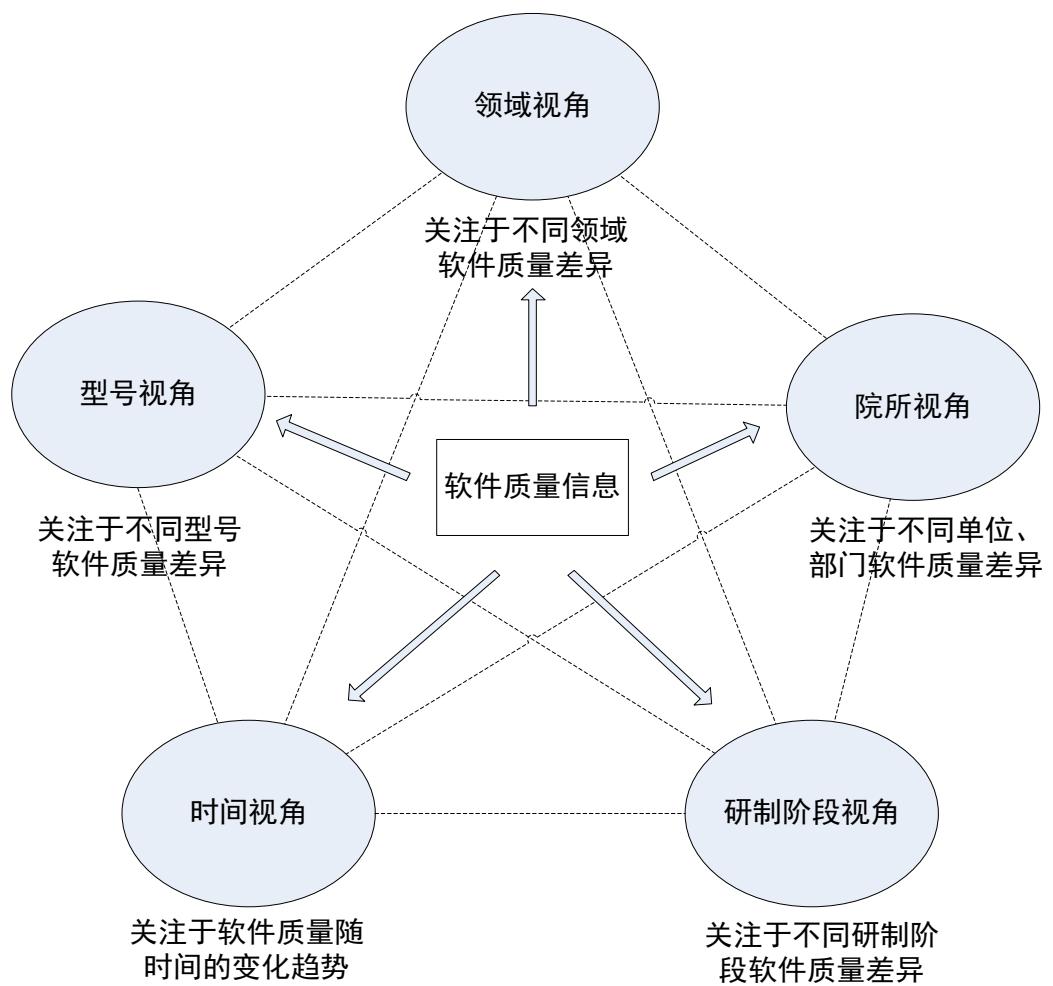


图 3-44 软件质量评估的几个重要视角

### (1) 时间视角的软件质量综合评估

时间视角下的软件质量评估关注于软件质量随时间的变化趋势。装备软件质量的评估不仅应对软件当前质量状况进行描述,也需要从时间角度对软件质量进行评价,观察各软件质量度量元随时间的变化趋势,为软件质量保证提供参考。如通过对历年千行代码问题数、千行代码关键/重要问题数、程序问题分类等质量度量元进行分析,可发现各软件质量度量元的改变特点,为今后的软件质量保障的要点与保障计划的制定提供支撑。例如下表中质量度量元 1 表示每年的软件可靠性水平,则通过基于时间视角的软件质量评估可直观说明软件可靠性逐年变化的趋势,为相关决策提供直观的数据支撑。

表 3-19 时间视角的软件质量数据示意

年份	质量度量元 1	质量度量元 2	.....	质量度量元 n
2018	0.97			
2018	0.98			
2017	0.95			
2016	0.92			
.....	.....			

### (2) 型号视角的软件质量综合评估

型号视角下的软件质量评估关注于不同型号软件质量的差异。软件质量的度量不仅需要评估各个质量度量元的整体水平，也需要从型号视角开展软件质量综合评估，评判不同型号的软件质量。通过不同型号软件质量的横向对比，可以在一定程度上避免好的特好，差的特差，型号软件质量整体平均值较好掩盖个别型号软件质量不达标的情况，便于发现软件质量水平较低的个别型号，并针对性的采取措施提高该类型号的软件质量水平。型号视角的软件质量数据如下表所示。

表 3-20 型号视角的软件质量数据示意

型号	质量度量元 1	质量度量元 2	.....	质量度量元 n
型号 A				
型号 B				
.....				

### (3) 领域视角的软件质量综合评估

领域视角下的软件质量评估关注于不同领域装备软件质量的差异。装备软件多种多样，分属于不同的领域，如飞控、雷达、指控等。不同领域软件的功能、性能、规模、实时性要求等千差万别，不同领域软件的软件质量也存在较大差异，因而从领域视角开展软件质量综合评估，考察分属不同领域装备软件的质量显得非常必要。领域视角的软件质量数据如下表所示。

表 3-21 领域视角的软件质量数据示意

领域	质量度量元 1	质量度量元 2	.....	质量度量元 n
领域 1				
领域 2				
.....				

#### (4) 院所视角的软件质量综合评估

院所视角下的软件质量评估关注于不同院所装备软件质量的差异。从院所视角开展软件质量评估,可以评判各装备研制单位装备软件质量的差异。通过不同院所装备软件质量的横向对比,便于发现各单位软件质量保障的强项与短板,为针对性的解决软件质量保障中存在的问题提供支撑。院所视角的软件质量数据如下表所示。

表 3-22 院所视角的软件质量数据示意

单位	质量度量元 1	质量度量元 2	.....	质量度量元 n
单位 1				
单位 2				
.....				

#### (5) 研制阶段视角的软件质量综合评估

研制阶段视角下的软件质量评估关注于不同研制阶段装备软件质量的差异。不同研制阶段的装备软件质量差异巨大,抽样、试样阶段软件存在问题众多、软件质量较差,而处于定型阶段软件质量则非常高,软件问题出现在研制初期与出现在定型阶段的重要程度显然不能同日而语。分不同研制阶段开展软件质量评估显得十分有必要。研制阶段视角的软件质量数据如下表所示。

表 3-23 研制阶段视角的软件质量数据示意

研制阶段	质量度量元 1	质量度量元 2	.....	质量度量元 n
抽样				
试样				

定型				
----	--	--	--	--

## 6. 复杂装备软件智能化与一体化测试平台

复杂装备软件智能化与一体化测试平台研究，旨在利用智能化手段辅助装备软件测试项目管理与测试实施，采用新技术、新手段改进传统测试模式，以提高测试工作效率，从而进一步保障装备质量。

### 6.1 平台建设方案

基于知识图谱的复杂装备软件智能化与一体化平台由以下三大模块、一个测试平台数据中心、一个测试知识图谱构成。

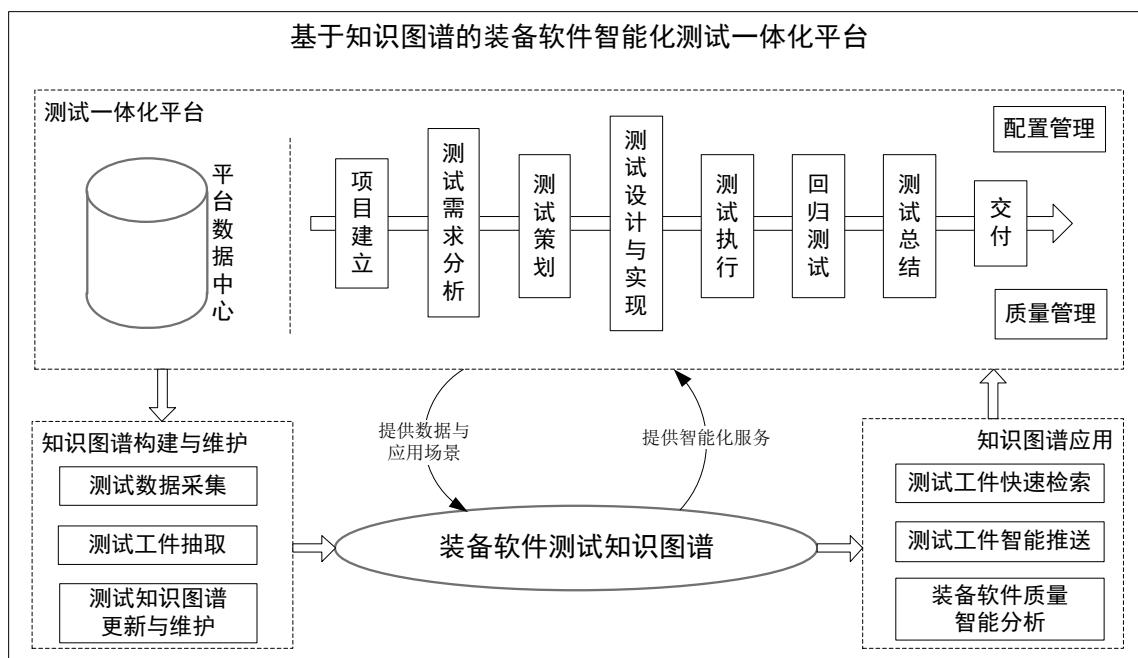


图 3-45 平台构成示意图

(1) 测试一体化平台：其实现了装备软件测试从项目建立到项目交付的全流程管理，符合相关国军标、行业标准对于测试过程管理、测试项目管理、质量管理与配置管理的要求，是整个基于知识图谱的复杂装备软件智能化与一体化平台项目建设的基础。测试一体化平台为装备软件测试知识图谱的构建提供数据，为基于知识图谱的智能化测试服务提供应用场景。

(2) 装备软件测试知识图谱构建与维护模块：主要包括测试数据采集、测试工件抽取与测试知识图谱更新与维护三个子模块。装备软件测试知识图谱构建与维护模块依托测试一体化平台，首先从测试过程、测试历史记录中采集测试数据，然后通过测试工件抽取与测试知识图谱更新与维护相关技术构建可扩展的装备软件测试知识图谱。

(3) 基于知识图谱的智能化服务：提供主要包括测试工件快速检索、测试工件智能推送与装备软件质量智能分析等智能化测试服务。基于装备软件测试知识图谱，深入装备软件评测各个阶段，从评测人员实际需求出发，提供测试工件检索、测试工件推送及软件质量智能分析等功能，以评测人员主动获取与平台主动推送相结合的方式提供基于知识图谱的智能化测试服务。

(4) 测试一体化平台数据中心：测试一体化平台数据中心，汇聚测试一体化平台中业务产生的以及平台外导入的测试相关的所有记录，是本项目开展智能化测试研究的数据基础。

(5) 装备软件测试知识图谱：装备软件测试知识图谱，以测试一体化平台数据中心的数据为基础，通过采集、筛选、解析形成体系化、精细化的测试知识图谱。装备软件测试知识图谱是对测试一体化平台数据中心中数据的凝练与提升。装备软件测试知识图谱是实现智能化测试的重要支撑。

## 6.2 平台技术架构

为保证基于知识图谱的复杂装备软件智能化与一体化平台的先进性、可扩展性与易集成性，拟在设计上采用基于面向服务架构实现基于知识图谱的复杂装备软件智能化与一体化平台的研制，对传统测试一体化平台进行智能化改造。

面向服务的架构（Service-Oriented Architecture，简称 SOA）是一种软件系统的架构，这种架构能通过连接完成某一特殊工作的独立功能

实现，也是为了解决在异构或同构环境下集成扩展的需求。SOA 是能将系统程序不同功能单元模块通过定义服务之间接口而联系起来的组件模型。它使用不偏向任何一方的方式定义接口，可以分别实现服务的硬件平台，操作系统和编程语言。这样有利于以一种统一和通用的方式在这样一个系统服务中进行交互。

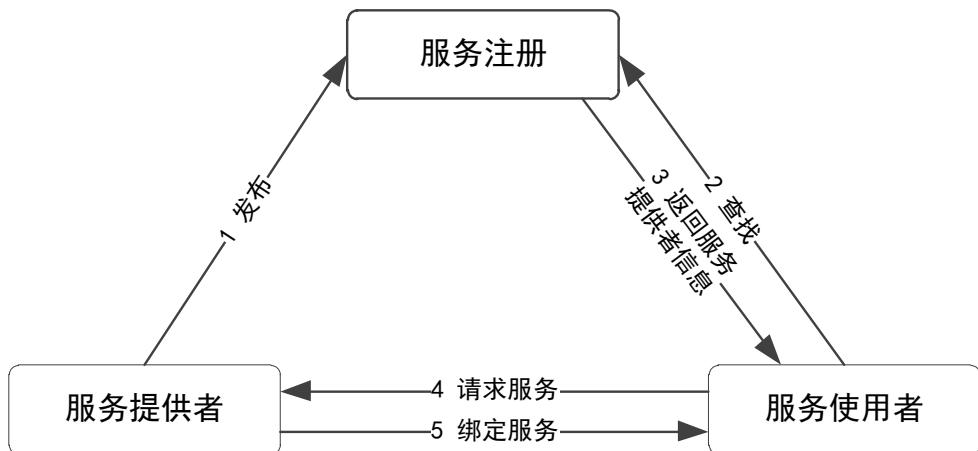


图 3-46 面向服务架构不同组件及其工作流程

### 6.3 智能化测试服务典型应用场景

通过对装备软件测试一般流程及流程中测试活动的梳理，可将智能化测试服务场景划分为快速检索、智能推送与智能评估三类。

#### (1) 快速检索

快速检索针对评测人员有确切目标或明确方向而主动发起的请求活动，如历史测试资产、相关测试工件的检索。在评测过程中，历史测试数据的重用与参考是提高评测人员工作效率与工作质量的重要环节。

#### (2) 智能推送

智能推送主要针对评测人员有大致范围，但仍需要调研分析才能得出结论，因而需要辅助信息提供决策参考的活动，如测试策略、测试方法、测试环境、测试组成员等的推荐。

#### (3) 智能评估

智能评估主要针对需要对大量数据进行整合分析从而给出结论的测试场景。如评测项目工作量、员工专业能力评估、知识图谱质量评估，装备软件质量评估等。

## （二）技术途径

### 1. 军用装备软件测试领域数据稀疏和跨任务场景的知识抽取技术

针对装备软件测试领域知识图谱构建过程的模型缺少训练数据和跨任务或跨应用场景知识抽取中标注和模型训练大等痛点问题，在上述知识抽取工作基础上，拟重点研究结合远程监督和迁移学习的知识抽取方法，分别解决训练数据集构建和跨领域数据标注的困难。

#### 1.1 基于远程监督的军用装备软件测试领域知识抽取

在军用装备软件测试知识抽取中，面对一个新的装备型号、新的测试任务、新的测试场景时，由于难以有效获取符合要求的领域数据集，往往需要人工构造相应的数据集来满足模型训练要求，然而这是一个耗时费力的过程，难以适应新型武器装备的快速迭代和反复试验的研制需求，因此成为军用装备软件测试知识抽取的一个难点问题。

首先，通过项目组已积累的领域词典，包括军用装备领域词典、军用软件开发词典、军用软件测试词典等领域词典以及军事百科词条等开源数据，自动对软件测试工件的数据集进行标注。与此同时，利用已构建的军用装备软件测试知识图谱，对数据集进行自动标注。在此远程监督的过程中，可以按照如下方法抑制噪声数据的产生：采用统计学习的方法对项目已经积累的标注语料进行分词统计分析，然后结合人工校验的方法，构造出一个面向语料信息的词典，该词典中只包括对于判定知识有用的词语；然后对自动标注而得的数据进行初步校验，只需要将标注数据进行分词拆分，然后扫描数据中包含的词语，如果该数据中不包含任何一个“有用”的词语，那么可以初步判定其为噪声，进行放弃。

然后，为了进一步保证标注数据结果的质量，通过投票机制过滤不可信的标注结果，并在后继模型训练中引入强化学习机制，奖励正确的标注，惩罚错误的标注，降低数据标签噪声，进一步提高标注结果质量。

在使用强化学习（记为 RL）的过程中，可以采用分层的 RL 进行知识抽取：一是高层的 RL 过程，即检测句子中关系指示符的高层次 RL 过程；二是低层的 RL 过程，即识别对应关系的相关实体的低层次 RL 过程。

### （1）高层 RL 的关系检测

**Option:** 对于处理的句子，在关系集合中选择满足的关系，当低层次的强化学习进入结束状态，agent 的控制将被高层次的 RL 接管去执行下一个 option。

**State:** 状态  $S$  由以下三者共同决定：当前的状态  $h_t$ ，最后一个 option 的关系类型向量  $v_t^r$  和上一个时间的状态  $s_{t-1}$ ，如下式所示：

$$S_t^h = f^h(W_s^h[h_t; v_t^r; s_{t-1}])$$

其中  $f^h$  是非线性变化， $h_t$  是 Bi-LSTM 得到的隐藏状态。

**Policy:** 关系检测的策略，也就是 option 的概率分布，如下式所示：

$$o_t \sim \mu(o_t | s_t^h) = \text{softmax}(W_\mu s_t^h)$$

**Reward:** 环境提供给 agent 的一个可量化的标量反馈信号，即 reward，计算方法如下式所示：

$$r_t^h = \begin{cases} -1, & \text{if } o_t \text{ not in } S \\ 0, & \text{if } o_t = NR \\ 1, & \text{if } o_t \text{ not in } S \end{cases}$$

### （2）低层 RL 的实体抽取

**Option:** 当高层的 RL 预测了一个非 NR 的关系，底层的 RL 会抽取其对应的实体。

**Action:** 在每个时间步上，action 会给当前的词分配一个 tag 进行标记。

**State:** 类似高层 RL 中的关系检测, 此处的状态计算方法如下式所示:

$$c_{t'} = g(W_h^l W s_{t'}^h)$$
$$s_t^l = f^l(W_s^l [h_t; v_t^e; s_{t-1}; c_{t'}])$$

其中,  $c_{t'}$  是上下文向量,  $h_t$  是当前词的隐藏状态,  $v_t^e$  是科学系的实体标签向量,  $s_{t-1}$  是上一阶段的状态,  $g$  和  $f$  都是多层感知机。

**Policy:** 由句子到实体的概率计算如下式所示:

$$a_t \sim \pi(a_t | s_t^l; o_{t'}) = \text{softmax}(W_\pi [o_{t'}] s_t^l)$$

**Reward:** 此处的奖励机制如下式所示:

$$r_t^l = \lambda(y_t) \cdot \text{sgn}(a_t^l = y_t(o_{t'}))$$

其中,  $\lambda(y)$  是用来降低非实体标签的权重, 定义如下式所示:

$$\lambda(y) = \begin{cases} 1, & \text{if } y \neq N \\ a, & \text{if } y = N \end{cases}$$

再次, 利用自动标注和标注清洗的数据, 训练知识抽取模型, 抽取得知识三元组。接着, 将可信的新发现知识三元组加入到训练数据集中, 通过自举迭代方式进行训练数据集的再次扩充和模型的训练, 直到新发现的知识三元组较少为止。

最后, 得到的知识图谱同样可以再次对新的数据集进行标注, 形成一个基于远程监督的测试数据自动标注、标注清洗、自举训练的迭代过程, 从而有效解决领域数据集稀缺和数据标注工作量大的军用装备软件测试领域知识图谱抽取难题。

## 1.2 基于迁移学习的军用装备软件测试领域跨任务知识抽取方法

针对以上军用装备软件测试知识抽取场景中面对跨型号、跨任务、跨场景测试中领域训练数据缺乏和数据稀疏的难题, 还可以进一步引入迁移学习技术, 在上述远程监督和自举迭代的基础上, 提高跨任务、跨型号、跨场景的模型训练效率。迁移学习可利用语料资源丰富的已完成源领域中学到的软件测试知识, 来辅助语料资源匮乏的目标领域上的学习任务。由

于神经网络通常使用基于梯度下降的方法进行增量训练，因此直接在源领域和目标领域之间使用梯度信息进行优化以实现知识转移是可行的。根据源领域和目标领域中数据的关系，即特征空间、类别空间、边缘概率分布、条件概率分布的异同，本方案采用的基于神经网络的迁移学习方法包括参数初始化方法（INIT）和多任务学习方法（MULT）。

迁移学习涉及领域（Domain）和任务（Task）两个重要概念，分别描述如下。领域 $\mathcal{D}$ 定义为由 $d$ 维特征空间 $X$ 和边缘概率分布 $P(\mathbf{x})$ 组成，即 $\mathcal{D} = \{X, P(\mathbf{x})\}$ ， $\mathbf{x} \in X$ 。给定领域 $\mathcal{D}$ ，任务 $\mathcal{T}$ 定义为由类别空间 $Y$ 和预测模型 $f(\mathbf{x})$ 组成，即 $\mathcal{T} = \{Y, f(\mathbf{x})\}$ ， $Y \in Y$ ，按统计观点预测模型 $f(\mathbf{x}) = P(Y|\mathbf{x})$ 解释为条件概率分布。

给定标注的辅助领域 $\mathcal{D}_s = \left\{ \left( \mathbf{x}_1^{(s)}, y_1^{(s)} \right), \dots, \left( \mathbf{x}_{n_s}^{(s)}, y_{n_s}^{(s)} \right) \right\}$ 和学习任务 $\mathcal{T}_s$ ，无标的目标领域 $\mathcal{D}_t = \left\{ \mathbf{x}_1^{(t)}, \dots, \mathbf{x}_{n_s}^{(t)} \right\}$ 和学习任务 $\mathcal{T}_t$ ，迁移学习的目标是在 $\mathcal{D}_s \neq \mathcal{D}_t$ 或 $\mathcal{T}_s \neq \mathcal{T}_t$ 条件下，降低目标领域预测模型 $f_t(\mathbf{x})$ 的泛化误差。根据特征空间、类别空间、边缘概率分布、条件概率分布的异同，迁移学习可以进一步分为多个子类。

(1)  $\mathcal{D}_s \neq \mathcal{D}_t$ : 源领域和目标领域的特征空间不同，例如，文档是用两种不同的语言写的。在自然语言处理的背景下，这通常被称为跨语言适应（cross-lingual adaptation）。

(2)  $P(\mathbf{x}_s) \neq P(\mathbf{x}_t)$ : 源领域和目标领域的边际概率分布不同，例如，两个文档有着不同的主题。这个情景通常被称为域适应（domain adaptation）。

(3)  $\mathcal{Y}_s \neq \mathcal{Y}_t$ : 两个任务的标签空间不同，例如，在目标任务中，文档需要被分配不同的标签。实际上因为不同的任务拥有不同的标签空间，但是拥有相同的条件概率分布，这是极其罕见的。

(4)  $P(Y_s|\mathbf{x}_s) \neq P(Y_t|\mathbf{x}_t)$ 。源任务和目标任务的条件概率分布不同，例如，源和目标文档在类别上是不均衡的。这种场景在实际中是比较常见的，诸如过采样、欠采样等方法被广泛应用。

参数初始化迁移方法首先在源领域上训练神经网络模型，然后直接使用训练得到的模型参数对目标领域的神经网络模型进行初始化，从而实现从源领域到目标领域的迁移。迁移后，可以将参数固定，即不对目标领域的神经网络模型进行训练。但是，当目标领域仍有部分标注数据时，可以利用目标领域的标注数据对模型参数进行微调（Fine-tune）以得到更好的效果。多任务学习迁移方法同时使用源领域数据和目标领域数据对一个神经网络模型进行训练，按 $\lambda: 1 - \lambda$ 的比例分别对源领域和目标领域的数据进行采样，模型的损失函数为：

$$\mathcal{L} = \lambda \mathcal{L}_S + (1 - \lambda) \mathcal{L}_T$$

其中， $\mathcal{L}_S$ 和 $\mathcal{L}_T$ 分别为模型在源领域和目标领域中的损失函数，且都经过了基于领域数据量大小的正则化， $\lambda \in (0, 1)$ 是为了平衡两个领域对最终结果的影响而引入的超参数。

将上述两种迁移方法运用在基于预训练的实体识别模型上，具体方法为：(1) 对于参数初始化迁移方法，首先，使用源领域训练数据对源领域模型进行训练，然后，用源领域模型中 LSTM 层的参数对目标领域的 LSTM 层进行初始化，之后可以选择两种策略利用目标领域的数据对目标领域模型进行训练：INIT-FineTune 和 INIT-Frozen，前者是对整个目标领域模型进行参数更新，后者只更新 CRF 层的参数。(2) 对于多任务学习迁移方法，首先，对源领域训练数据和目标领域训练数据进行随机采样，以超参数 $\lambda$ 为样本来自于源领域数据的概率，然后同时对源领域模型和目标领域模型进行训练，两个模型在训练过程中使用相同的 LSTM 层参数。这种方法在每构建一个新的目标领域模型时，都将重新训练源领域模型，当源领域训练数据规模较大时会产生较高的时间成本。

## 2. 军用装备软件测试领域知识图谱的高效和高质量融合技术

大规模软件测试领域知识图谱高效和高质量融合技术，分别从四个方面进行实现：基于语义文档相似度的软件测试本体匹配方法、基于相似度传播模型的软件测试本体匹配算法、大规模软件测试知识图谱实例匹配的空间复杂度处理、基于锚点的规模软件测试知识图谱实例匹配算法。

### 2.1 基于语义文档相似度的软件测试本体匹配方法

本课题首先将软件测试本体元素的相关文本按照其对元素语义的描述方式进行组织和划分，并称由此构造出的文本为元素的语义描述文档。为了避免引入与元素关联不大的其它文本，语义描述文档构造过程被严格限制在元素的语义子图内，即语义描述文档是基于语义子图的。此外，语义描述文档不考虑本体语言中的元语，如`rdfs:Class`和`owl:hasValue`等。构造过程还包括文本预处理，即进行词干提取和频繁词去除等。

首先，本体中的概念、关系和实例均有一个基本描述文档，它由元素局部名(`local name`)、标签(`label`)和注释信息构成。元素 $e$ 的基本描述文档表示如下：

$$DS_{base}(e) = \lambda_1 * W_{localName} + \lambda_2 * W_{label} + \lambda_3 * W_{comment} + \lambda_4 * W_{otherAnotation}$$

其中 $W_{localName}$ 表示元素的局部名称， $W_{label}$ 表示`rdfs:label`标签文本， $W_{comment}$ 表示`rdfs:comment`注释文本， $W_{otherAnotation}$ 表示其它的注释文本；每种文本的权重 $\lambda_i$ 取区间[0,1]上的值。公式中的文本和最终的描述文档均是由带权重的词构成的集合。 $+$ 表示集合间的合并操作。这里的文本不包含本体语言中的元语，如`rdfs:Class`和`owl:hasValue`等。

由于本体在之前的预处理中经过丰富处理，因此通过诸如`owl:equivalentClass`等公理声明的等价元素间的基本描述是相同的。

本课题希望寻找到合理的方式来组织软件测试中的概念和关系的相关文本，并利用这些文本来作为判断匹配的线索。为达到这个目的，这里

从两个方面进行考虑：一方面，根据概念和关系的语义描述方式对文本进行重新组织，这种组织方式具有通用性；另一方面，为了避免得到的文本中包含那些与元素并不太相关以及不太重要的文本信息，将描述文本构造的范围局限于语义子图内。将描述文档转化为向量空间之后，两文档之间的相似度可以用其对应的向量之间的夹角余弦来表示，即文档之间的相似度可表示为对应向量的余弦。

## 2.2 基于相似度传播模型的软件测试本体匹配算法

本体图由三元组构成，其上的相似度传播应同时考虑三元组中的边和点。如果将相似流算法中的传播条件简单扩展到三元组上，即如果三元组间对应位置上的一对元素相似，则相似度会传播到其它两对元素上，但是该传播条件存在明显不足：首先，它会产生大量的候选相似点对，并生成规模庞大的对偶连接图；其次，这种传播条件会产生大量错误的候选相似对。本课题针对本体三元组的特点，提出了三元组相似度传播强条件，简称强条件。

给定本体中的任意两个三元组  $t_i = \langle s_i, p_i, o_i \rangle$  和  $t_j = \langle s_j, p_j, o_j \rangle$ ， $S_s$ 、 $S_p$  和  $S_o$  分别表示三个对应位置上的元素相似度。当且仅当  $t_i$  和  $t_j$  均满足下列三个条件时，它们之间才存在相似度传播： $S_s$ 、 $S_p$  和  $S_o$  中至少有 2 个的相似度值大于阀值  $q$ ；如果  $t_i$  中包含本体元语，则  $t_j$  的对应位置也必须是相同的本体元语； $t_i$  或  $t_j$  中包含的元语不能超过 2 个。

第一个条件保证传播得到的相似度具有一定可信度且两三元组采用相同元语陈述事实。第二个条件将本体中的元素定义和声明排除于相似度传播之外，因为由这类三元组得到传播结果可信度较低。

在一次相似度传播后，三元组中某一对元素相似的增加量为另外两对元素的相似度的乘积，即  $S_s^i = S_s^{i-1} + S_p^{i-1} \times S_o^{i-1}$ 。

假设相似矩阵最多经过  $L$  次相似度传播后达到收敛。由于在强相似传播条件下，每次传播可能使得对偶连接图中的最大路径增长 1，并设图中

的最大路径长度为  $D$ ，则对偶连接图的更新不可能超过  $D$  次。设相似度传播中固定点计算最大收敛次数为  $M$ ，由于每次需要计算对偶连接图中所有相似对的传播结果，如果对偶连接图的平均大小为  $N$ ，那么整个相似度传播算法的时间复杂度为  $O(L * D * M * N)$ 。

### 2.3 大规模软件测试知识图谱实例匹配的空间复杂度处理

空间复杂度问题是大规模软件测试知识图谱融合中不能忽视的挑战。本课题通过精心组织关键数据结构，合理分配和回收内存，并借助诸如数据库等工具，以克服空间复杂度问题。

本课题主要采用基于图数据的大规模实例存储和解析。由于大规模实例数据无法像传统实例匹配系统一次将数据读入内存并进行随后一系列解析操作，所以需要特定存储手段，并保证高效读取和查询等解析操作。大规模实例是一种典型的图数据，可以将其存储在图数据库（如 neo4j）中。然而，虽然图数据库在一定程度上解决了数据的存储问题，但却并不提供实例匹配过程中常用的各种解析操作，如遍历特定类型的元素或获取元素邻居信息等，这将限制匹配效率的提高。因此，需要在图数据存储的基础上提供高效的常用数据解析操作。

本方案设计一系列基于哈希索引的实例匹配中主要解析操作的高效实现。首先，以实例元素（概念、关系和实例）为单位，分别以其 URI 为唯一键值构造相应的 RDF 声明哈希表，即将每个元素直接的语义信息都建立索引，这能保证所有元素的直接语义信息都能实时获取。其次，将  $k$  ( $2 < k < 5$ ) 个语义相邻元素对应的三元组作为一个存储单位，通过 Hash 索引方式进行存储，从而保证局部区域元素可以实时获取，便于利用空间相邻原则。最后，对链接数据实例，利用编辑距离等基础文本相似度算法，对链接数据中的大量实例进行局部敏感哈希索引 (Locality-Sensitive Hashing, LSH)，这种索引能保证具有一定相似的实例能作为一个集合被快速查询到，同时局部敏感哈希理论保证了不相似实例被该集合遗漏的

可能性很小。这三种索引技术能为匹配算法提供高效的支持，并便于在其基础上实现实例的分块或匹配锚点的预测。

## 2.4 基于约简锚点的大规模软件测试知识图谱匹配算法

大规模实例匹配中普遍存在两种事实：(a) 大规模实例包含大量由 is-a 和 part-of 关系构成的层次结构，正确的匹配不能破坏这种层次结构。(b) 大规模实例间的元素匹配具有区域性特点，即实例  $O_1$  的特定区域  $D_i$  中的元素大多会被匹配到实例  $O_2$  的特定区域  $D_j$  中，以块为单位的实例匹配结果也证实了该事实。这两种事实为寻找有效的匹配方法提供了新思路。首先，由于匹配不能破坏原实例的结构层次，当能够确定  $O_1$  中的概念  $A$  与  $O_2$  中的概念  $B$  匹配，则  $A$  的子概念不必再与  $B$  的父概念做匹配计算，从而能减少很多无谓的相似度计算。其次，由于匹配的元素集具有区域性，则可认为元素及其邻居通常只与另一实例中的部分元素相关，而与其它大多数元素都无关，因此，当能确定  $A$  与  $B$  不匹配时，就可以认为  $A$  的邻居与  $B$  也不会匹配，这同样能避免很多无谓的相似度计算。

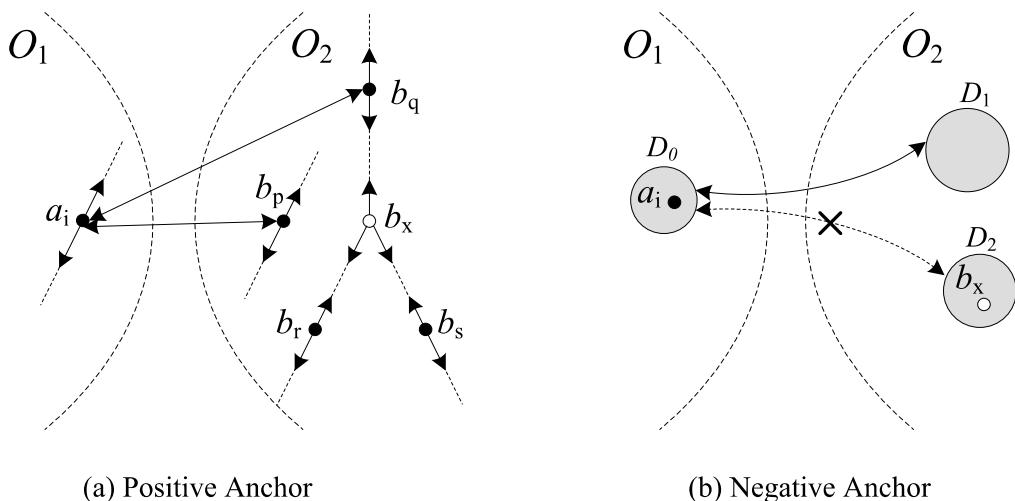


图 3-47 正锚点和负锚点

### 2.4.1 基于正锚点的大规模实例匹配算法

基于上面的分析，由于匹配不能破坏原实例结构层次的原则，因此当能够确定  $O_1$  中的概念  $A$  与  $O_2$  中的概念  $B$  匹配，则  $A$  的子(父)概念不必再与  $B$

的父(子)概念做匹配计算，从而能减少很多无谓的相似度计算。本课题将根据这种思想得到的能跳过后继匹配位置的概念称为正锚点，具体定义如下：

给定  $O_1$  中概念  $a_i$ ，设它与  $O_2$  中的元素  $b_1, b_2, \dots, b_n$  的相似度为  $Si_1, Si_2, \dots, Si_n$ ，称相似度大于阀值  $ptValue$  的  $O_2$  中的概念构成的集合为  $a_i$  对应的正锚点(Positive Anchor)，即  $PA(a_i) = \{b_j | Si_j > ptValue\}$ 。

上图中的(a)显示了实例层次结构与匹配。如果在计算  $a_i$  的相似度时，发现它与  $b_p$  或  $b_q$  具有较高的相似度，因此有理由相信这样的结论： $a_i$  与  $b_p$  或  $b_q$  匹配的可能性较大，这样带来的直接好处是：在随后的相似度计算中，可以直接跳过  $a_i$  的子概念与  $b_p$  (或  $b_q$ ) 的父概念，以及  $a_i$  的父概念与  $b_p$  (或  $b_q$ ) 的子概念的相似度计算。本项目称满足这种特点的  $b_p$  和  $b_q$  为  $a_i$  的正锚点。

当计算  $O_1$  中概念  $a_i$  与  $O_2$  中全部概念的相似度后，便可以根据  $a_i$  的正锚点来预测后继匹配计算中可跳过的位置，称这种根据正锚点预测得到的匹配位置为正约简集。显然，正约简集只有在  $a_i$  的正锚点中的概念集合不为空时才可能得到，正锚点中的概念可能不只一个，为保证正约简集中包含较多正确的可跳过位置，预测时取相似度最大的 top-k 个正锚点。此外，计算匹配过程的匹配顺序将影响到最终获得的约简集的大小，因此算法中还需要寻找最佳的匹配顺序选择方法，并对此进行严格证明。

#### 2.4.2 基于负锚点的大规模实例匹配算法

大规模实例在匹配结果上还存在这样的事实：大规模实例间的元素匹配具有区域性特点，即实例  $O_1$  的特定区域  $D_i$  中的元素大多会被匹配到实例  $O_2$  的特定区域  $D_j$  中，这称为匹配的区域性特点。由此可认为元素及其邻居通常只与另一实例中的部分元素相关，而与其它大多数元素都无关，即当能确定  $A$  与  $B$  不匹配时，就可以认为  $A$  的邻居与  $B$  也不会匹配，这同样

能避免很多无谓的相似度计算。本课题将根据这种思想得到的能跳过后继匹配位置的概念称为负锚点，具体定义如下：

给定  $O_1$  中概念  $a_i$ ，设它与  $O_2$  中的元素  $b_1, b_2, \dots, b_n$  的相似度为  $Si_1, Si_2, \dots, Si_n$ ，称相似度小于阀值  $ntValue$  的  $O_2$  中的元素构成的集合为  $a_i$  对应的负锚点 (Negative Anchor)，即  $NA(a_i) = \{b_j | Si_j < ntValue\}$ 。

上图中的(b) 显示了大实例匹配的区域性特点，其中实线双箭头表示区域  $D_0$  与  $D_1$  中的元素匹配，虚线双箭头表示  $D_0$  与  $D_2$  不匹配。设  $a_i$  和  $b_x$  分别是  $D_0$  与  $D_2$  中的元素，则有理由相信  $a_i$  和  $b_x$  的相似度较小，由此可进一步推测  $a_i$  的邻居与  $b_x$  的相似度同样较小，于是在随后的相似度计算中，可以直接跳过  $a_i$  的邻居与  $b_x$  的相似度计算。称满足这种特点的  $b_x$  是  $a_i$  的负锚点。

根据负锚点的定义， $a_i$  和它的负锚点具有较低的相似度，即  $a_i$  很可能与负锚点在语义上无关，因此可进一步推测  $a_i$  的邻居与  $a_i$  的负锚点同样不相关，这样  $a_i$  的邻居在做相似度计算时可跳过  $a_i$  的负锚点中包含的元素。这里的邻居不限于直接邻居，而包含在概念层次结构中并与  $a_i$  距离为  $nScale$  的概念。对于由此而产生的可跳过的匹配位置的集合，称为负约简集。在基于负约简集构造的匹配算法中，需要注意两个问题：首先，负锚点的无限制传播将会导致负约简集的可信度降低，因此需要对负约简集的传播进行限制；其次，元素相似度计算次序也将影响负约简集大小，因此还需要寻找产生最大负约简集的有效方法。

正锚点和负锚点提供了两种提高大规模匹配效率的手段。利用这两种锚点，相似度计算过程中可跳过大多数位置的计算，从而降低计算的时间复杂度。显然，正锚点和负锚点无法事先确定，因此本项目在相似度计算中动态确定锚点，并利用得到的锚点预测后继相似度计算中那些可直接跳过的位置。项目将着重研究两种基于锚点的匹配预测算法，然后再进一步探讨如何结合两种锚点算法。

### 3. 基于知识图谱的测试用例可解释性推荐技术

测试用例推荐是测试工件推荐的重要内容。相似软件需求对应的测试用例具有一定的相关性，因此，可计算与当前软件需求相似的历史软件需求，从知识图谱中获取相似需求对应的测试用例作为推荐。如何从海量测试用例中，快速找寻合适的测试用例是本项目的关键技术，同时需要给出一定的辅助信息，帮助测试人员理解推荐这些测试用例的原因。此外，仅有少数的测试用例产生了问题单，在相关的测试用例基础上，推送给测试人员易发现缺陷的测试用例也是需要考虑的问题，这样可引导测试人员快速发现、不遗漏软件问题。

本项目提出基于知识图谱的测试用例可解释性推荐方法，基于测试项、软件信息从知识图谱中找寻可复用测试用例集合，推荐更易发现缺陷的测试用例，提高软件测试的缺陷发现率，保证软件测试的质量。测试用例由许多关键词（中断、通讯、周期时间等）组成，不同关键词情况下出现问题的概率不同，因此需要首先依据关键词信息，找寻软件测试知识图谱中历史测试用例，然后根据关键词组合对测试用例进行排序，将易出问题的测试用例优先推荐，从而使推荐具有可解释性。

下图为一个新抽取的测试项 A，包含了“测试内容及测试要求”、“测试约束条件”、“测试方法”和“判定准则”。

```
{  
    "测试内容及测试要求": "每30ms能正确响应一次485通信中断，每5ms能正确响应一次数据采集中断",  
    "测试约束条件": "正确连接各测试设备和通信接口模拟器",  
    "测试方法": "使用插桩的方法。通过上位机模拟器与下位机的连接与否可以控制下位机工作在RS485  
        通信连接，在程序中设置内部定时器测试RS485通信连接时485中断的间隔时间和数据采  
        集中断的间隔时间",  
    "判断准则": "485通信中断时间间隔为30±0.1ms，数据采集中断间隔时间为5±0.1 ms"  
}
```

图 3-48 测试项示例

对上述测试项，可以抽取为如下的测试项 A 局部图：

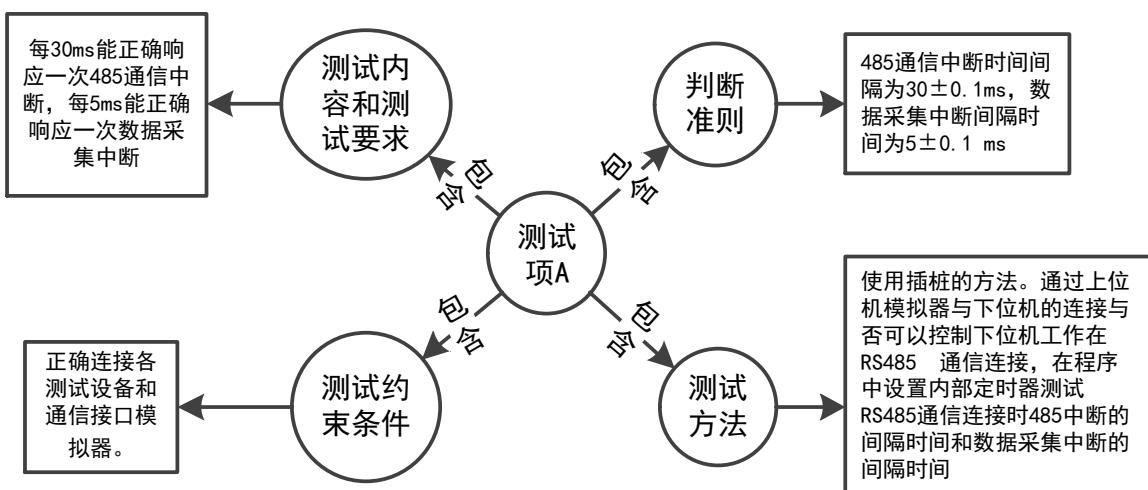


图 3-49 测试项 A 局部图

其次，测试项 A 所属软件信息(软件型号、软件类型、软件所述信息、编程语言、软件需求等)也是在知识图谱中查询测试项的关键信息。在具体的知识查询中，首先依据测试项目 A 的上述四项描述信息，以及测试项 A 所述的软件信息，检索测试知识图谱中与测试项 A 相似的测试项，可找寻出知识图谱中多个测试相项。如下图所示，测试项 B 和测试项 C 是与测试项 A 相似的测试项，即考虑了软件信息，又考虑了与测试项相关的“测试约束条件”、“测试方法”等，需要采用局部图匹配方式找寻相似测试项，体现了知识图谱的优点。若不采用知识图谱，仅仅采用关键词匹配方式，很难将“软件->需求->测试项->测试用例”等知识链条考虑进来，从而会降低后续测试用例的推荐准确率。其次，对于最终推荐的测试仪用例，将包含上述知识链表的知识图谱子图返回给测试人员，可大大增加测试用例推荐的可解释性，辅助测试人员快速理解测试用例分解原因。

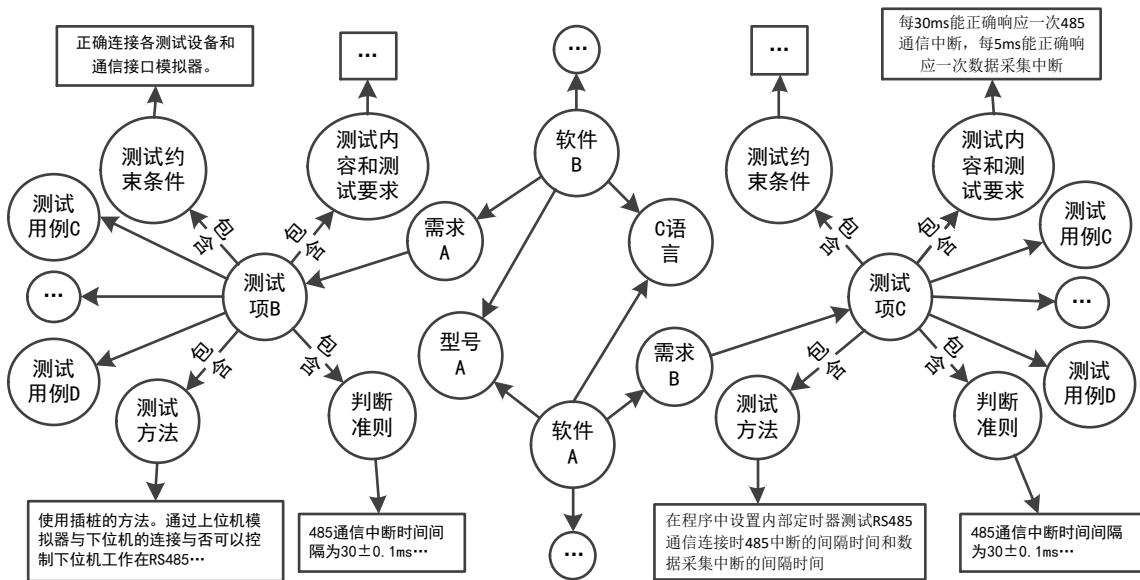


图 3-50 相似测试项局部知识图谱

在找寻到相似测试项后，可获取候选测试用例集合，如何对测试用例评价，筛选最终推荐测试用例是本项目接下来的重点。本项目拟提出一种新的基于关键词发现的推荐模型，从候选测试用例集合中筛选出最有可能产生问题单的测试用例。

首先，整理已有的测试项、测试用例和对应的问题单数据，构建样本集；然后，采用命名实体识别模型识别出测试项和测试用例中的关键词信息，如“中断”，“数据采集”。接着，采用信息增益计算每一个关键词对“是否产生问题单”这一类别的信息增益。最后，选择信息增益大的关键词作为特征，对样本集的每条测试项生成向量表示（每个关键词或关键词组合对应向量中的一维），然后通过有监督的方式来训练分类器。使用训练好的分类器，能够预测出每条测试项产生缺陷的概率，可以为用户优先推荐容易产生缺陷的测试项的测试用例，从而帮助测试人员设计出更易发现缺陷的测试用例。此外，由于本方法计算出了用于分类的关键特征，在推荐测试用例时，可输出测试项、测试用例和问题单的局部图，此外标记出推荐测试用例的关键词信息，以方便测试人员理解。

本项目提出的模型总体框架如下图所示：

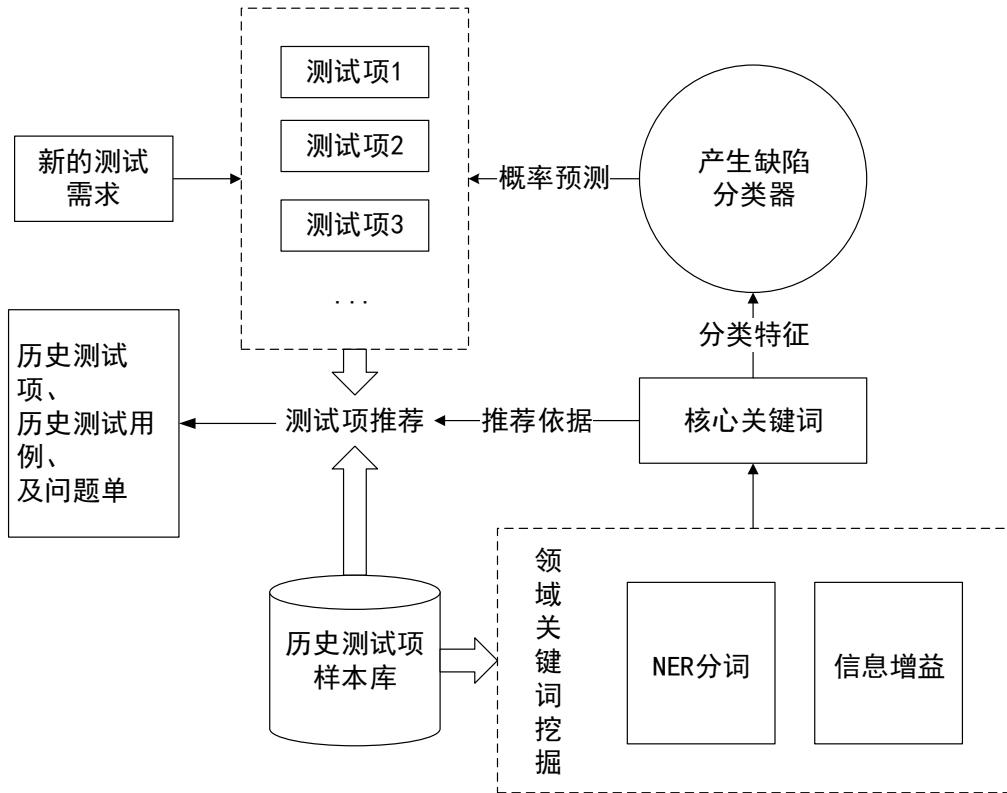


图 3-51 总体模型框架

### 步骤 1：基于命名实体识别模型的领域分词

本项目采用基于深度学习的命名实体识别模型，将实体识别问题转化为序列标注问题，将一个句子中的每个字标记成 BIEO 四种标签（B 表示词的开始，I 表示词的中间部分，E 表示词的结尾，O 表示不重要的词），模型输入是字符序列，输出是一个标注序列，因此这是一个标准的端到端序列标注问题。本项目采用 NER 模型对搜集的测试项、测试用例的描述进行领域分词，通过 NER 模型，去除对语义表达没有意义的词语，从而提取句子中的关键信息（命名实体），然后输入到信息增益模型，计算核心关键词。

### 步骤 3：信息增益提取核心关键词

分类特征的选择在于选取对训练数据具有分类能力的特征，而信息增益就是描述特征的分类能力的一种准则。

信息增益引入了熵的概念，表示“一个系统的混乱程度”。系统的不确定性越高，熵就越大。假设集合中的变量  $X=\{x_1, x_2 \dots, x_n\}$ ，它对应在集合的概率分别是  $P=\{p_1, p_2 \dots, p_n\}$ 。那么这个集合的熵  $H(X)$  表示为：

$$H(X) = - \sum_{i=1}^n p_i \log_2 p_i$$

特征 A 对训练集 D 的信息增益  $G(D, A)$ ，等于集合 D 的经验熵  $H(D)$  和特征 A 给定条件下 D 的条件经验熵  $H(D|A)$  之差，即

$$G(D, A) = H(D) - H(D|A)$$

本项目将经过 NER 模型分词后获得的领域词汇看作是特征，通过对其信息增益的计算，得到其对整体数据集分类（会不会产生缺陷）的重要程度，从而就可以筛选出对推荐有解释意义的一些核心关键词。

### 步骤 3：分类模型训练

根据步骤二中得到的核心关键词，使用词袋模型对训练集进行编码，将每个句子变成向量  $X = (x_1, x_2, \dots, x_{n-1}, x_n)$ ， $n$  为核心关键词的数量，输入到分类模型的网络中进行训练，分类算法常用的有很多种，本项目使用朴素贝叶斯分类器。

### 步骤 4：测试用例的推荐

测试用例的推荐流程如下图：

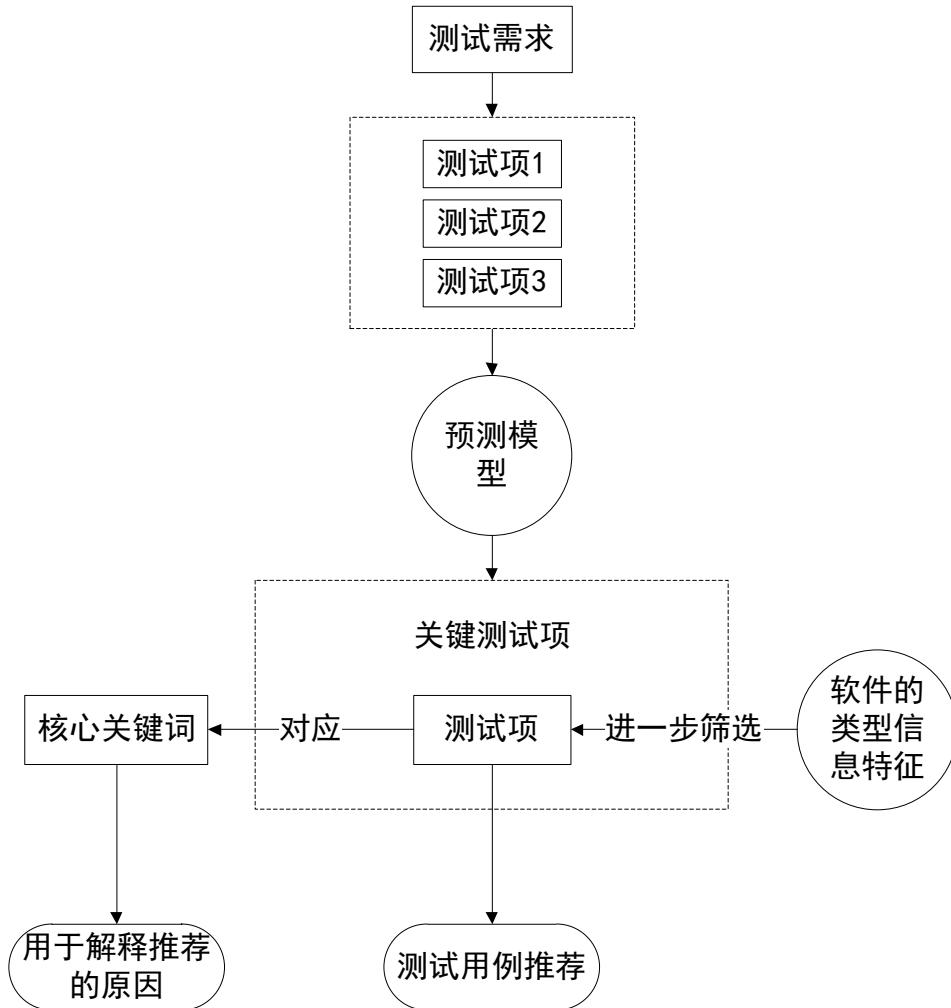


图 3-52 测试用例推荐流程

将每条候选测试用例输入步骤三中训练好的分类模型，预测出每个测试用例出现缺陷的概率，为用户优先推荐出现缺陷概率高的关键测试用例，并将其对应的核心关键词返回用户，向用户说明容易产生缺陷的原因。

#### 步骤五：测试用例的筛选

接着，使用被测软件的软件类型、编程语言、型号、运行环境等静态信息从知识图谱中进一步筛选出符合条件的测试用例，最终将“软件->需求->测试项->测试用例”等知识链条返回给测试人员，帮助测试人员设计测试用例，从而提高软件测试的效率和质量。

### (三) 创新点

#### 1 基于机器学习的软件设计说明和代码匹配技术

面向装备软件代码的测试工件提取与融合,针对软件设计说明与代码的匹配问题,创新性提出基于自然语言处理的设计说明条目化,进而采用语义分析完成设计说明条目化到结构化的转化,同时,研究基于切片的代码过滤技术,过滤无关代码,提取核心代码片段;最后标记少量数据集训练机器学习模型,实现软件设计说明和代码的匹配,服务面向代码的测试工件提取与融合。

#### 2 结合语义子图和约简锚点对知识图谱进行高质量和高效融合

面向装备软件测试领域多源异构数据构建的软件测试知识图谱,针对本体层异构和实例层异构问题,创新性提出基于语义子图和约简锚点对知识图谱中进行高质量和高效融合。首先,提出一种基于语义子图的高质量本体匹配方法,实现高质量的本体层融合;然后,针对大规模实例匹配面临时间复杂度和空间复杂度的挑战,提出一种基于约简锚点的实例匹配算法,确定待融合知识图谱实例层中的正负锚点,构造实例的约简集合,降低传统知识融合方法的时间复杂度和空间复杂度,实现高效的实例层融合。

#### 3 采用远程监督和迁移学习解决数据稀缺和跨任务的领域知识抽取

针对装备软件测试领域知识图谱构建过程中缺少训练数据和跨任务、跨应用场景知识抽取中标注数据和模型训练工作量大等痛点问题,创新性提出采用远程监督和迁移学习解决领域知识抽取中的数据稀缺和跨任务问题。首先,在领域知识抽取模型训练中引入远程监督技术,通过领域词典、开源百科数据、已有知识图谱自动进行数据标注,形成模型训练数据集,并利用强化学习降低标注数据噪声,从而解决模型训练数据稀缺的难题。然后,在跨任务、跨应用和跨领域军用装备测试场景中引入迁移学习技术,将已有知识抽取模型的训练数据或模型迁移到新的测试场景中,有

效减少数据标注工作量和模型训练时间，提高知识抽取效率，保障新型武器装备快速迭代研制的需求。

#### 4 采用知识图谱实现测试用例可解释性推荐

军用装备软件的相似需求对应的测试用例具有一定的重叠性，历史测试用例可辅助测试人员快速完成测试项分析。本项目创新性地提出基于知识图谱的测试用例可解释性推荐技术，采用命名实体识别模型分析测试项、测试用例等数据中关键实体信息，获取候选测试用例集合；将测试用例表示为不同命名实体组合，结合“测试用例->问题单”关联信息训练测试用例缺陷预测模型，挖掘更易发现问题的测试项；最后结合知识图谱中“需求->测试项->测试用例->问题单”知识链条，为用户推荐包含测试用例的知识图谱子图，并给出关键命名实体组合信息，实现测试用例可解释性推荐。

### 四、研究进度、成果形式及应用方向

#### (一) 研究进度

表 4-1 进度安排

年 度	年度目标	年度研究内容	年度成果形式
2019 年	完成项目论证	(1) 完成项目建议书撰写。	
2020 年	完成基于装备软件评测流程的测试工件定义与分析研究、完成测试工件自动提取与融合技术研究、完成软件测试知识图谱自动化构建技术研究、开展知识快速检索与智能推送及测试用例生成技术研究	(1) 完成装备软件评测通用流程分析研究； (2) 完成测试工件定义与分析研究； (3) 完成面向武器装备测试流程数据的清洗技术研究； (4) 完成面向武器装备软件测试文档的测试工件提取与融合技术研究； (5) 完成面向武器装备软件代码	研究报告： (1) 基于武器装备软件评测流程的测试工件定义与分析研究报告； (2) 测试工件自动提取与融合技术研究报告；

	<p>的测试工件提取与融合技术研究；</p> <p>(6) 完成基于本体构建方法的软件测试领域本体构建技术研究；</p> <p>(7) 完成多策略军用装备软件测试知识三元组抽取技术研究；</p> <p>(8) 完成基于语义子图和约简锚点的大规模测试知识图谱融合技术研究；</p> <p>(9) 完成基于最小代价的知识动态更新技术研究；</p> <p>(10) 完成知识图谱质量评估技术研究；</p> <p>(11) 完成知识图谱语义检索技术研究；</p> <p>(12) 完成基于知识图谱的测试工件智能推送技术研究。</p>	<p>(3) 软件测试知识图谱自动化构建技术研究报告。</p> <p>知识产权：</p> <p>论文 2 篇；</p> <p>专利 2 项。</p>	
2021 年	<p>完成知识快速检索与智能推送及测试用例生成技术研究、完成典型装备领域软件质量评价技术研究、完成复杂装备软件智能化与一体化测试平台研制</p>	<p>(1) 完成基于模板的测试用例自动生成技术研究；</p> <p>(2) 完成装备软件质量度量元分析技术研究；</p> <p>(3) 完成装备软件质量度量元归一化技术研究；</p> <p>(4) 完成多视角装备软件质量评价技术研究；</p> <p>(5) 完成复杂装备软件智能化与一体化测试平台建设相关技术研究。</p>	<p>研究报告：</p> <p>(1) 知识快速检索与智能推送及测试用例生成技术研究报告；</p> <p>(2) 典型装备软件质量评价技术研究报告；</p> <p>(3) 复杂装备软件智能化与一体化测试平台研究报告；</p>

			<p>(4) 基于知识图谱的复杂装备软件智能化测试技术总结报告。</p> <p>知识产权：</p> <p>论文 2 篇；</p> <p>专利 1 项；</p> <p>软件著作权 1 项。</p> <p>原型软件：</p> <p>基于知识图谱的复杂装备软件智能化与一体化测试原型系统。</p>
--	--	--	---

## （二）项目成果

### 1. 技术文件类成果

#### （1）技术总结报告

基于知识图谱的复杂装备软件智能化测试技术总结报告。

#### （2）专题报告

基于武器装备软件评测流程的测试工件定义与分析研究报告；

测试工件自动提取与融合技术研究报告；

软件测试知识图谱自动化构建技术研究报告；

知识快速检索与智能推送及测试用例生成技术研究报告；

典型装备软件质量评价技术研究报告；

复杂装备软件智能化与一体化测试平台研究报告。

(4) 专利和著作权

拟申请专利 3 项；

申报软件著作权 1 项。

(5) 论文

拟撰写论文 4 篇。

## 2. 实物类成果

本项目的主要实物类成果是基于知识图谱的复杂装备软件智能化与一体化测试原型系统。该系统将历史测试知识充分融合利用，构建软件测试知识图谱，辅助提高测试工作质量和效率，充分保障装备质量。

主要功能：

(1) 支持从非结构化装备软件数据中识别领域标签和特定实体，如战技指标、软件类型等；支持从非结构化装备软件数据中抽取知识三元组；

(2) 装备软件领域命名实体识别准确率不低于 90%；

(3) 面向非结构化数据的知识抽取的召回率不低于 85%，准确率不低于 85%；

(4) 装备软件测试知识图谱具有可扩展性和自学习性，实体数量不低于 20 万，关系数量不低于 60 万，覆盖飞控、指控、测发控、雷达、导引头等典型装备软件类型；

(5) 测试过程数据的自动化收集与分析，支持知识图谱的动态更新；

(6) 实现基于测试过程数据对装备软件进行质量评价，支持装备软件质量多角度评估与分析；

(7) 推送测试工件类型不少于 10 种；

(8) 智能推送 TOP3 准确率不低于 90%；

(9) 典型装备软件质量评价指标不少于 3 种，基于测试数据的装备软件质量评价准确率不低于 80%；

(10) 支持面向文档的测试工件提取与融合；

(11) 支持面向软件代码的测试工件提取与融合。

验证环境：

基于知识图谱的复杂装备软件智能化与一体化测试平台，平台采用 B/S 架构，将平台应用部署于应用服务器，即可通过浏览器请求对应页面的方式对相关功能进行验证。

指标验证：

在“二、研究目标、研究内容与技术指标”中的“技术指标”章节详细说明了技术指标验证方法与条件。

### **(三) 应用方向**

本项目主要成果是基于知识图谱的复杂装备软件智能化与一体化测试平台原型系统，能够针对测试过程中的重要测试活动提供快速检索、智能推荐与智能评估等智能化测试服务，将极大改变传统的软件测试模式，同时也能极大提高软件测试的效率。本项目成果能够直接应用于复杂装备软件评测等专业评测领域，也能够较快捷地扩展至数据丰富且具有智能化应用需求的其他领域。

(1) 为装备软件专业第三方评测机构提供智能化测试解决方案

随着软件规模、软件型号任务数量的快速增长，软件大规模快速交付给软件质量保证提出了新的挑战。面临越来越大的装备软件评测工作量，越来越紧张的项目周期，以及越来越高的软件质量要求，传统的软件测试方法已不能满足当前装备软件质量保证的需求。基于知识图谱的复杂装备软件智能化与一体化测试平台以当前装备软件评测的一般过程为基础，能够支持测试项、测试用例、测试问题单等十余种典型测试工件的快速检索与智能推送，在测试过程中为评测人员及评测管理人员提供全方位的智能化辅助，提高软件测试的效率，从而更好地保障装备软件质量。

(2) 为装备软件测试动态管理提供有力支撑

由于第三方测试机构是在项目后期介入，需求文档可能相对滞后，且缺少后续开发过程中多次需求变更文档，造成测试方在对软件需求的把握上无法做到全面和到位，这对软件测试的过程有一定影响。此外，第三方测试通常时间较短，在这种情况下，由于缺乏高效第三方测试管理工具，造成测试团队花费在测试管理的精力相对较多。装备软件测试知识图谱的天然可扩展性能够基于需求变更快速实现知识图谱的动态更新，辅助测试方快速把握需求变更及其影响，使测试人员专注于测试本身，从而提高测试人员效率，提高测试质量。

(3) 借助装备软件测试知识图谱能够深入挖掘软件缺陷的形成机理，为软件研制单位的软件设计与研制阶段规避软件缺陷提供指导。

研制单位在委托第三方测试时，往往是在软件开发的系统集成和验收阶段。装备软件测试知识与经验往往难以对研制方软件的需求、设计和编码阶段产生影响。借助装备软件测试知识图谱进行关联关系挖掘与分析，能够快速发现常见软件缺陷位置及其形成机理，为软件研制单位的软件设计与研制阶段规避软件缺陷提供指导。

(4) 提供领域知识图谱构建及应用整体解决方案

基于知识图谱的复杂装备软件智能化与一体化测试平台是典型领域基于知识图谱的智能化实践。其以历史评测过程数据为基础构建装备软件测试领域知识图谱，贯穿数据处理、领域知识图谱构建、领域知识图谱应用全过程，形成一套完整的领域知识图谱构建与应用技术链，能够为数据资源丰富且有意愿采用发挥历史数据潜在价值的相关单位提供领域知识图谱构建及应用整体解决方案。

## 五、经费概算

### (一) 按研究内容分解经费

本课题总经费 170 万元，按研究内容分解如下表所示。根据研究内容分工情况，北京计算机技术及应用研究所预算经费小计 110 万元，南京大

学预算经费 30 万元，东南大学预算经费 30 万元。

表 5-1 按研究内容分解概算（万元）

序号	研究内容	经费分解	承研单位
1	智能化软件测评流程分析与数据收集技术	30	北京计算机技术及应用研究所
2	面向软件文档及代码的测试工件自动提取与融合技术	30	南京大学
3	软件测试知识图谱自动化构建技术	30	东南大学
4	知识快速检索智能推送与测试用例生成技术	30	北京计算机技术及应用研究所
5	典型装备软件质量评价技术	30	北京计算机技术及应用研究所
6	基于知识图谱的复杂装备软件智能化与一体化测试平台	20	北京计算机技术及应用研究所
合计		170	——

## （二）按年度分解经费

表 5-2 年度经费概算

经费类别	年度	2020 年	2021 年	合计
材料费		24.44	8.1	32.54
专用费		0	0	0
外协费		14	15	29
燃料动力费		0	0	0
事务费		14.86	7.62	22.48
固定资产折旧费		4	3.93	7.93
管理费		14.8	6.93	21.73
工资及劳务费		36.3	20.02	56.32
收益		0	0	0

经费类别\年度	2020 年	2021 年	合计
不可预见费	0	0	0
合计	108.4	61.6	170

## 六、研究条件及保障措施

### (一) 本单位研究基础

#### 1. 北京计算机技术及应用研究所

北京计算机技术及应用研究所是我国最早从事计算机研究的大型骨干专业研究所之一，是装发军用计算机及软件技术专业组成员单位，是装发军用计算机研制生产定点单位。航天软件评测中心成立于1996年6月，是国内最早成立的从事军用高可靠高安全软件检测的专业机构，挂靠在北京计算机技术及应用研究所。航天软件评测中心是国防科工局批复建设的国防科技工业软件测试和评价实验室，国内第一家通过总装军用软件测评实验室认可的评测机构，同时也是国家高端装备和智能制造软件质量监督检验中心，具有国家实验室（CNAS）、国防实验室（DILAC）、资质认定（CMA）等资质，具备载人航天工程、探月工程软件评测资质。成立近20年来创造了多项国内第一，是国内高可靠高安全软件检测领域成立时间最早、规模最大、能力最全、服务用户最广的专业软件检测机构。

航天软件评测中心成立以来，已圆满完成了以载人航天、探月工程为代表的国家重大科技工程，以高新工程为代表的各军兵种国防装备，为国防现代化和国民经济建设做出了重要贡献，先后获得“全国五一劳动奖状”和“全国三八红旗集体”，国家人事部、总装备部和国防科工委联合授予的“中国载人航天工程突出贡献集体”，以及高新工程中的多项突出贡献奖。

经过20年发展，北京计算机技术及应用研究所在以下方面具有研究基础。

#### (1) 具备的相关资质

国防科技工业软件测试与评价实验室  
国家高端装备和智能制造软件质量监督检验中心  
中国合格评定国家认可委员会实验室认可资质（CNAS）  
军用校准和测试实验室资质  
国防科技工业实验室资质（DILAC）  
检验检测机构资质（CMA）  
载人航天工程软件第三方评测资质  
探月工程软件评测资质  
空军装备研制环境与可靠性试验和软件测试定点实验室资质  
空间科学卫星工程软件测评机构资质  
国防科技工业自主可控计算机软硬件适配技术创新中心  
工信部安全可靠联合攻关平台  
国防科技工业网络安全创新中心  
航天科工网络完全与自主可控技术研发中心

## （2）软件测评能力

航天软件评测中心在军用软件实验室能力认可中具备嵌入式软件 4 种级别、19 种测试类型能力，非嵌入式软件的 3 种级别、19 种测试类型能力，是总装军用软件测评实验室目录中实验室中能力最全的实验室。

航天软件评测中心主持或参与制定了 14 份国家军用标准、行业标准；近年针对面向对象技术、FPGA 建立了相应的标准，填补了国内行业空白。航天软件评测中心从事航空航天和国防装备软件检测 20 年，在高可靠高安全领域积累了丰富的技术和经验，具备将标准体系推广至其它高端装备领域的技术储备和能力。

## （3）人员规模优势

航天软件评测中心现有员工 130 余人，其中研究员 13 人、高工 40 人，80%以上具有硕士学位，是国内规模最大的软件检测机构。拥有一个

联合博士培养点、四个硕士学位授予点，有 6 人在科工局、总装、载人航天工程和探月工程担任软件专家。航天软件评测中心建成了一支既深入了解高可靠高安全背景又掌握软件专业知识的高水平专业队伍。

#### （4）完善的质量管理体系

航天软件评测中心制定了完善的质量管理体系，质量体系覆盖 GJB2725A《测试实验室和校准实验室通用要求》、《军用软件测评实验室测评过程和技术能力要求》、CNAS CL01《检测和校准实验室能力认可准则》、DILAC AC01《检测实验室和校准实验室能力认可准则》相关要求。质量体系包含质量手册 1 份，程序文件 30 份，作业文件 61 份，保证软件测试工作质量受控。

#### （5）科研能力

航天软件评测中心在软件可靠性安全性、软件评测、软件工程等专业领域研究方面处于国内领先或先进水平；十一五、十二五、十三五期间承担了多项国防科工局、总装的科研项目，累计经费 9000 余万元。

承担了国防科工局的《软件复杂度度量与评估》、《基于缺陷模式的软件可靠性分析方法》、《武器装备嵌入式软件综合测试平台技术研究》等课题研究。

承担了总装备部的《基于全数字仿真的软件可靠性测试环境》、《实时软件测试充分性技术研究》、《嵌入式软件全数字虚拟测试环境技术》、《军用软件测试与评估技术》、《武器装备嵌入式软件模型及构件化软件开发技术研究》等课题的研究。

承担了载人航天办公室的《软件安全性标准和技术发展趋势研究》、《载人航天工程软件面向对象开发方法研究》课题研究。

航天软件评测中心通过这些课题的研究，紧跟了国际工业和学术领域在高端装备软件的安全性可靠性发展趋势、软件检测和认证方法。

#### （6）在人工智能和知识图谱技术研究基础

① 初步构建了航天软件测试知识图谱。航天软件评测中心自 1996 年成立以来，承接了来源于航空、航天、核、陆军、火箭军、载人航天、探月工程等多个型号的装备软件的测试任务，积累了飞控、指控、测发控、雷达、导引头等多个领域测试数据产品，为了将二十多年的历史积累有机整合，航天软件评测中心构建了航天测试知识图谱，并基于此实现了快速检索、智能推荐等初步应用。

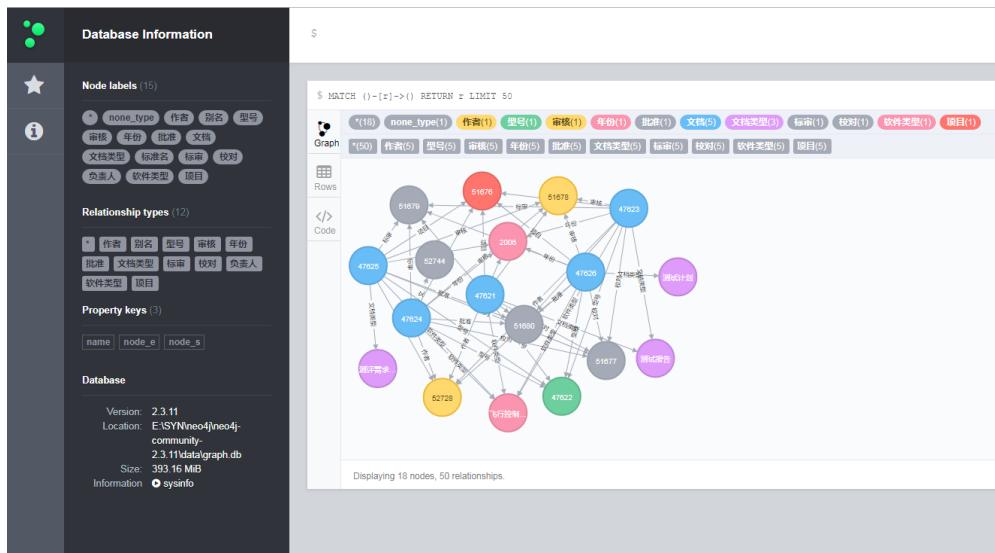


图 6-1 软件测试知识图谱示意图

② 基于航天软件测试知识图谱初步实现了测试检索工具 TestLinker。基于 TestLinker 提供的领域搜索功能，评测人员能够基于关键字快速找到需要参考的历史测试项、测试用例与相关测试报告等，在一定程度上加速了历史知识重用，提高了测试人员的工作效率。



图 6-2 TestLinker 知识检索示意图

③ 初步实现了测试项推荐系统，测试项推荐系统依据需求文档，识别出相似需求，进而将测试项和测试用例推送给测试人员，可辅助测试人员快速完成测试需求分析、测试项分解。

序号	测试项标识	测试项名称	测试项内容	设计日期	详情
0	██████████	检████	检████	██████████	
1	██████████	检████	检████	██████████	
2	██████████	周████	周████	██████████	
3	██████████	任████	任████	██████████	
4	██████████	任████	任████	██████████	
5	██████████	系████	系████	██████████	
6	██████████	处████	处████	██████████	
7	██████████	任████	任████	██████████	
8	██████████	任████	任████	██████████	
9	██████████	任████	任████	██████████	

图 6-3 测试项推荐系统

### (7) 平台研发能力

北京计算机技术及应用研究所研发了软件测评实验室过程管理系统 STLIMS，能够支持航天软件评测中心装备软件测试全流程项目管理与流程管理。目前，软件测评实验室过程管理系统（STLIMS）已在航天软件评测中心应用多年，成为支撑航天软件评测中心测试顺利实施的重要基础平台。

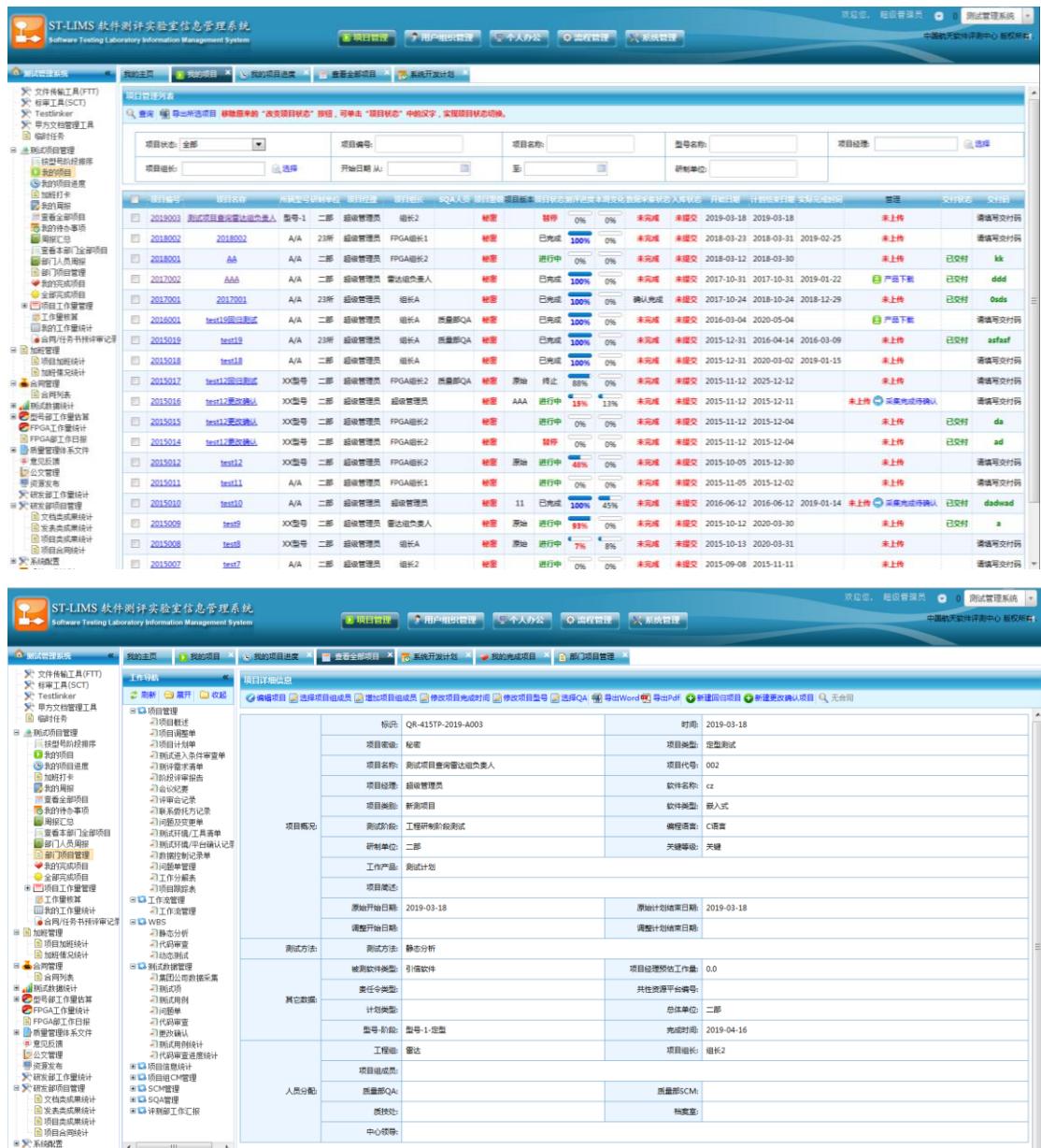


图 6-4 软件测评实验室过程管理系统 STLIMS

## 2. 东南大学

东南大学是中央直管、教育部直属的全国重点大学，是“985工程”、“211工程”和“双一流”A类重点建设的大学之一。经过一百多年的创业发展，如今的东南大学已成为一所以工科为主要特色，多学科协调发展的综合性、研究型大学。东南大学计算机科学与工程学院是国家特色专业建设点和江苏省品牌专业建设点，学院拥有“计算机科学与技术”一级学科博士点，包括“计算机系统结构”、“计算机软件与理论”、“计算机应用

技术”全部 3 个二级学科博士点，以及“图像处理与科学可视化”博士点，计算机科学技术专业入选“双一流”建设学科，计算机应用技术连续被评选为国家重点学科，是国内具有明确特色和较高学术水平的计算机院系。学院拥有一支结构合理、充满朝气、以中青年教师为主的 100 多人规模的高水平师资队伍，建有计算机网络和信息集成教育部重点实验室、江苏省计算机网络技术重点实验室、江苏省网络与信息安全重点实验室、中法生物医学信息研究中心，以及国家 863/CIMS“网络与数据库”工程实验室、中国教育科研网华东（北）地区网络中心等一批科研实验基地，近年来，承担了国家重点研发计划、国家自然科学基金重点项目等各类科研项目 300 多项，发表学术论文 3000 余篇，科研成果获国家科技进步二等奖 1 项、教育部自然科学二等奖 1 项、江苏省科学技术二等奖 2 项、山西省自然科学二等奖 1 项；授权发明专利 135 项，牵头制定国家标准 1 项。

东南大学参研团队是国际最早开展知识图谱研究的团队之一，在知识图谱的研究和应用中取得了大量国际前沿的研究成果。在知识图谱基础理论方面，首次提出语义子图概念用于精确描述本体元素语义；基于语义子图提出一种通用的本体匹配框架，较好地解决了通用本体匹配问题；提出一种有效的相似度传播方法处理缺乏充足或规则文本信息的本体匹配问题；最早独立提出在本体映射中进行映射调试的必要性，并给出一种的映射调试启发式算法，可提高映射结果的精度；在大规模本体匹配中，首次提出一种无需对大规模本体进行划分的高效匹配方法。项目团队在知识图谱构建、知识推理、知识融合、大数据分析与挖掘等方面进行了深入研究，已在 IJCAI, AAAI, WWW, VLDB, ICDE, ISWC, Science China 等权威会议和期刊发表了 200 多篇高水平学术论文，参与研制本体映射系统 Lily、语义搜索系统 Faclon-S 以及第一个大规模中文知识图谱 Zhishi.me 等经典工作。

项目团队前期参研 2017 年“十三五”全军共用信息系统装备预研项目装研发项目《基于知识图谱的 XXX 技术》，参研国家重点研发计划《村镇聚落空间重构数字化模拟及评价模型》。两项课题的研究内容均涉及知识图谱，和本课题存在一定的相关性。此外，项目团队已完成相关的国家和省部级研究基金 20 多项，包括 2011 年国家自然科学基金《面向语义 Web 复杂应用环境的本体映射关键技术研究》，2014 年中国博士后科学基金一等项目《大规模知识融合的高效处理方法研究》，2015 年国家自然科学基金《本体匹配中的参数和策略调谐问题研究》，2015 年 863 课题《开放域知识集成、推理与检索关键技术及系统》。

综上，团队成员对本课题相关技术非常熟悉，具有丰富的实践应用经验，积累的这些经验将能指导新项目的顺利开展。

一些代表性的应用成果如下：

(1) 基于知识图谱的科技情报分析系统：从开源科技情报数据源自动收集 2 亿篇学术论文、1.2 亿学者信息、8000 多万专利，在此基础上构建科技情报知识图谱，并实现领域专家挖掘、专家画像、技术趋势预测等分析功能，该系统已在多家国防和事业单位得到推广应用，如下图所示。

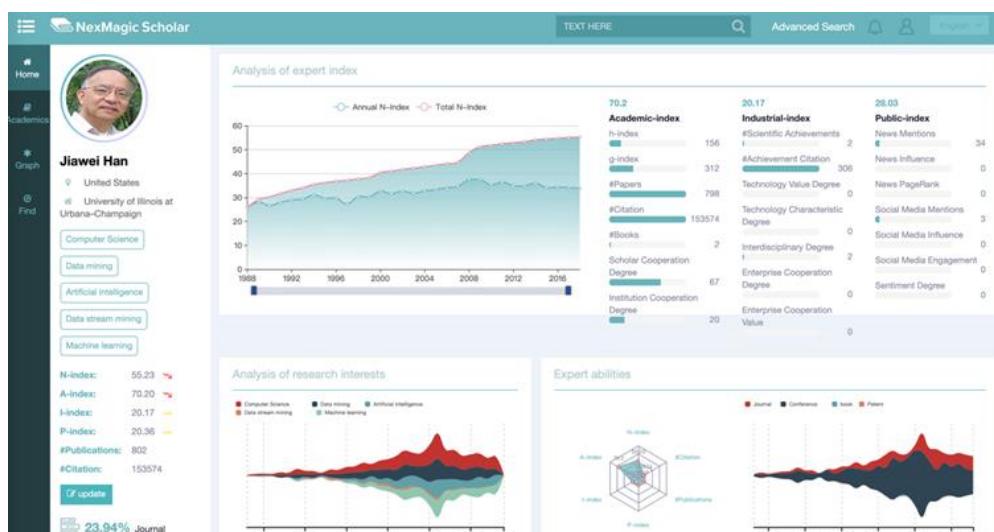


图 6-5 基于知识图谱的科技情报分析系统

(2) 基于知识图谱的公共安全智能问答和语义搜索系统：利用知识图谱集成百亿级的人、案、车、出行、通信、住宿、工作等公共安全相关数据，在该知识图谱上实现基于知识图谱的智能问答和语义搜索，辅助办案查人等公共安全决策，该系统已在多地公安机构推广应用，如下图所示。



图 6-6 基于知识图谱的公共安全智能问答和语义搜索系统

(3) 基于知识图谱的汽车行业情报分析和智能决策系统：收集汽车行业论坛、新闻、社交、销售和政策等海量数据，抽取知识构建汽车行业知识图谱，进一步量化车型指标体系，形成对汽车的实时多维度多指标分析，辅助汽车厂商的生产和销售决策，该系统已在上汽集团推广应用，如下图所示。

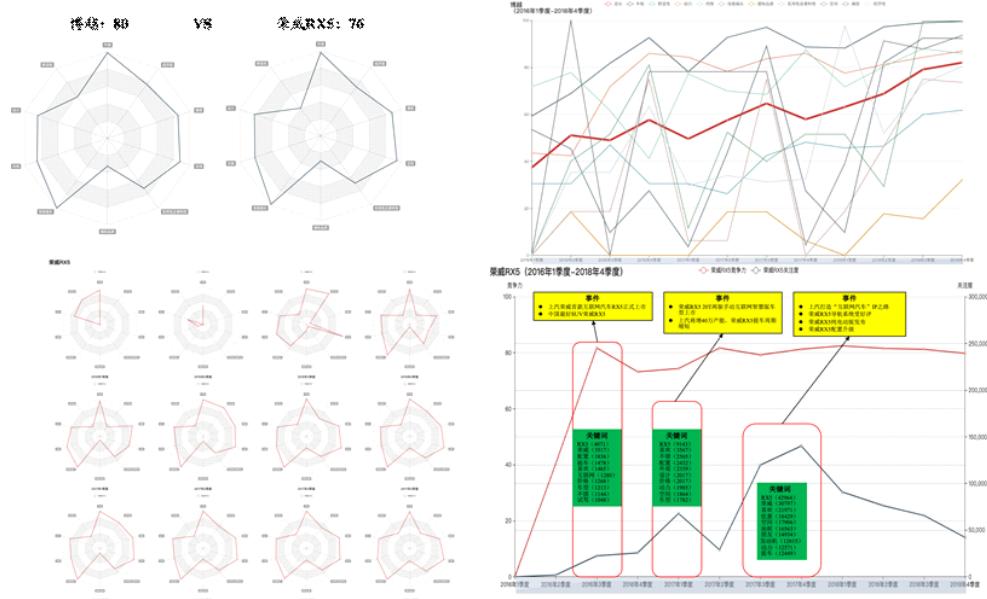


图 6-7 基于知识图谱的汽车行业情报分析和智能决策系统

(4) 基于知识图谱的智能制造生产管理决策系统：针对高端装备型号复杂的生产管理过程构建领域本体，自动抽取生产数据构建知识图谱，通过语义查询、时空分析、语义关联挖掘，辅助生产管理过程中的智能决策，该系统已在三一重工和某重点型号推广应用，如下图所示。



图 6-8 基于知识图谱的汽车行业情报分析和智能决策系统

### 3. 南京大学

南京大学是一所历史悠久、声誉卓著的百年名校，国家首批“211 工程”和“985 工程”重点建设高校。南京大学计算机科学与技术系拥有国家一级重点学科和江苏省重中之重学科，拥有计算机科学与技术一级学科博士学位授予权，近年来在队伍建设、人才培养、科学研究等方面一直位居国内先进行列。项目参与团队所依托的南京大学计算机软件新技术国家重点实验室连续在 2007 年和 2012 年两次评估中，获得信息科学领域优秀类国家重点实验室。实验室具有良好的项目研究硬件支撑条件；在新型程序设计与软件方法学、分布计算与并行处理、系统软件及信息安全以及智能化与多媒体软件支撑技术研究方面具有丰富的研究经验和研究成果。实验室共承担国家“973”计划、“863”计划、国际合作与横向协作项目等 560 余项。在重要国际期刊和重要国际会发表论文 5000 余篇，出版学

术著作 47 部，获国家和部省级奖励 79 项。本实验室积极拓宽科技成果的应用与转化渠道，通过与企业合作等方式，在促进国家信息化建设，推动软件产业发展方面又取得良好进展，取得了较好的经济和社会效益。实验室现有固定研究人员 57 人，已初步形成了一支素质良好、朝气蓬勃、结构合理的研究队伍，拥有雄厚的师资力量。

## （二）人才条件

柯文俊：高级工程师，目前主要从事软件系统航天领域模型驱动开发、知识图谱、深度学习等人工智能方向技术的开展及管理工作。

“十三五”期间组织负责了国防基础科研《航天领域模型驱动软件设计开发技术》、《软件复杂度度量与评估技术》的研究工作，在软件工程、软件理论方面积累了丰富的项目经验。

2014 年起，主持构建了“航天软件测试知识图谱”，在知识库、特定领域知识图谱构建方面积累了丰富经验。

2015 年至今先后主导了《软件密集系统故障预测方法》、《JS 语言应用集成验证技术》、《复杂软件失效表征规律及在线失效预测方法研究》等装备发展部基金及预研项目，《基于大数据分析的电子文件智能定解密技术》的国防基础科研项目，在自然语言处理、知识检索与推理技术方面具有深入的研究及完善的项目管理能力，在国内学术刊物和学术会议上发表了 10 余有关人工智能方向相关论文。

表 6-1 主要参研人员情况

姓名	出生年月	职务 (职称)	单位	分工	年工作量	备注
柯文俊	1990. 03	高级工程师	北京计算机技术及应用研究所	负责人	8	
汪鹏	1977. 10	副教授	东南大学	技术研究	6	

姓名	出生年月	职务 (职称)	单位	分工	年工作量	备注
何铁科	1988. 02	助理研究员	南京大学	技术研究	6	
付修锋	1985. 03	高级工程师	北京计算机技术及应用研究所	技术研究	6	
陈静	1987. 04	高级工程师	北京计算机技术及应用研究所	技术研究	6	
李伟	1986. 02	高级工程师	北京计算机技术及应用研究所	技术研究	6	
陈旭	1993. 12	工程师	北京计算机技术及应用研究所	软件开发	6	
王坤龙	1990. 06	工程师	北京计算机技术及应用研究所	软件开发	6	
杨雨婷	1992. 03	工程师	北京计算机技术及应用研究所	技术研究	6	
宋颖毅	1995. 02	助理工	北京计算	软件开发	8	

姓名	出生年月	职务 (职称)	单位	分工	年工作量	备注
		工程师	机技术及应用研究所			
周晓宇	1974. 08	副教授	东南大学	技术研究	6	
张柏礼	1973. 1	副教授	东南大学	技术研究	6	
张祥	1979. 5	副教授	东南大学	技术研究	6	
崇志宏	1972. 2	副教授	东南大学	技术研究	6	
朱超	1990. 1	博士生	东南大学	软件开发	5	
张良	1989. 4	博士生	东南大学	软件开发	5	
徐忠锴	1996. 2	硕士生	东南大学	软件开发	5	
刘欢	1994. 4	硕士生	东南大学	软件开发	5	
陈振宇	1978. 11	教授	南京大学	技术研究	4	
邹智鹏	1996. 03	硕士生	南京大学	软件开发	6	
沈思媛	1996. 09	硕士生	南京大学	软件开发	6	
袁为	1997. 06	硕士生	南京大学	软件开发	6	
汪汇	1997. 04	硕士生	南京大学	软件开发	6	
扬奇	1992. 10	硕士生	南京大学	软件开发	6	
程雷	1997. 07	硕士生	南京大学	软件开发	6	

### (三) 研究条件

北京计算机技术及应用研究所、南京大学和东南大学联合申请本项目，各单位分工明确，结合各自优势，可以为本项目提供良好的研究条件。

具体研究条件说明如下：

#### 1. 项目团队具有良好的实验环境与基础设施

主承研单位北京计算机技术及应用研究所下属的航天软件评测中心，是总装军用软件测评实验室目录中实验室中能力最全的实验室。现有面积 5000 平米，现有软件测试工具 36 套，FPGA 工具 12 套，仿真测试环境 22

套；十三五期间新建 87 台设备，将建成基于云的静态检测平台、虚拟化测试系统、导弹武器系统级测试、信息安全检测、FPGA 检测条件，具备 13 万亿次计算能力高性能计算集群，累计投入技改经费 2 亿元，是国内软件检测条件最为完善的机构。

北京计算机技术及应用研究所具备测试一体化平台建设从设计开发到应用的环境试验条件及相关的软硬件开发环境，以及较强的系统设计及软件研制能力；同时，北京计算机技术及应用研究所具备大型系统的综合测评能力。因此，项目团队能够为复杂装备软件智能化与一体化测试平台的研制提供良好的软硬件设施与验证环境，并全程进行功能与性能测试。

## **2. 项目团队拥有丰富的装备软件测评数据积累**

主承研单位北京计算机技术及应用研究所下属的航天软件评测中心，是国内最早的武装装备软件第三方评测机构，拥有二十多年的技术及评测过程数据积累，涵盖飞控、雷达、测发控、指控等多个专业领域，可为本项目提供充足的测试相关数据，能够为智能化软件测评平台的应用提供良好的实验场景。

## **3. 项目团队拥有成熟的支撑工具和技术积累**

项目团队拥有成熟的支撑工具和技术积累，课题团队一直在从事软件测试、知识图谱及其交叉领域的相关研究工作，是国际较早开展知识图谱研究的团队，取得了优异的理论研究成果，发表在国际顶级学术期刊及会议 ICSE、FSE、ASE、TSE、IJCAI、WWW、VLDB、ICDE 等。申请了测试及知识图谱相关发明专利 20 余项，部分研究成果已经在百度、阿里、华为等一流软件企业应用。项目团队参与研制了本体映射系统 Lily、语义搜索系统 Faclon-s 以及第一个大规模中文知识图谱 Zhishi.me 等极具影响力的系统和工程项目。项目团队大部分成员曾经参加过团队的相关科研项目，或者合作发表过软件测试、知识图谱、数据挖掘等方面的论文，积累了丰

富的研究经验。团队成员在众包测试、知识图谱和数据挖掘领域都有坚实的研究基础和技术积累。

#### (四) 外协条件

本项目的外协单位为灵玖中科软件(北京)有限公司，该公司拥有业内领先的大数据语义检索的能力，为本项目知识快速检索与推送提供强有力的技术支撑。

灵玖中科软件(北京)有限公司多年专注于面向海量异构互联网信息，研究网络大数据搜索、自然语言处理、社会计算与信息安全等关键技术，以自然语言理解为主要手段进行情报挖掘。

灵玖中科软件(北京)有限公司的大数据搜索与挖掘技术已经应用于全球二十余万家机构，包括国家统计局、中国证监会、中国邮政集团、国家气象局、国家新闻办公室、最高人民法院等国家单位，中国上市公司协会、中国对外承包工程商会、北京市园林局、解放军某部等事业与机关，以及海航集团、中国网、富基融通 eFuture、四维图新、缔元信等大中型企业。代表项目成果如下：

##### (1) 国家电网：通过语义深度理解，构建电力领域专业知识库。

大数据搜索与挖掘的技术，对电力领域数十万本专业书籍和国家电网报近十年的文本语料进行深入的文本和语义的分析挖掘，整合了垂直领域知识，构建了专业领域的知识库，从而促使电网传媒在传播过程中更专、更准、更广、更及时。

##### (2) 西城公安：重构公安大数据资源，提高打击犯罪效率

公安智能语义分析平台是一个集文本挖掘、快速检索、高维可视和数据预测等功能于一体的警情数据智能化管理与分析平台。大数据搜索与挖掘的最新技术，通过对公安沉淀的海量高价值密度的非结构化数据的分析挖掘，全面梳理和整合公安大数据资源，打造出具有数据分析处理、数据挖掘能力的公安智能语义分析平台，实现公安大数据价值的挖掘与利用。

## **(五) 管理保障**

北京计算机技术及应用研究所长期从事装发的预先研究工作，有一套完善的预研项目管理制度和办法。北京计算机技术及应用研究所领导高度重视预研工作，制订了一套完善的预研工作管理制度。所级有主管副所长专门负责，计划管理部门有专门的主管领导和调度主抓计划的安排、落实和课题协调，并设有预研专家组，负责预研项目的技术把关和审查。

计划部门统一组织编写和下发预研项目年度计划，按照计划定期检查并考核，针对各预研项目进展情况，不定期召开预研工作调度会，对出现的各种问题及时协调解决。对于关键性节点加强考核并组织专家统一把关，以保证项目的顺利进行。

领导严格指定人员分工、落实负责制，明确项目组成员的研究任务及考核指标，协调本项目与其他项目之间的关系；课题组组长负责课题的统筹规划，组织人员针对本项目研究中的技术难题进行技术攻关；并定期召开课题研究工作例会，加强项目组成员之间的沟通与协调，确保本项目的完成质量与进度；研究资金专款专用，确保本项目有充足的经费支持。