

**新闻文本分类**

**功能测试方案文档**

项目名称： 分类8新闻文本分类

申 请 者： 刘兴瑞 王方 张潇潇 王亚涛

指导教师： 代祖华

学 院： 计算机科学与工程学院

目录

[一.测试方案的目的 2](#_Toc25832)

[二.测试方案与计划的区别 2](#_Toc409)

[1、定义不同 2](#_Toc30533)

[1、层次不同 2](#_Toc30533)

[三.测试方案 2](#_Toc17050)

[1、测试需求分析 2](#_Toc30533)

[2、测试内容 3](#_Toc2610)

[3、测试进度计划 3](#_Toc12305)

[4、风险管理 3](#_Toc3630)

[5、质量 3](#_Toc3630)

[四.测试策略 4](#_Toc409)

[1.测试类型 4](#_Toc18246)

[2.两者比较 4](#_Toc31459)

[3.测试方式不同 5](#_Toc25804)

[4.测试目的不同 6](#_Toc8382)

[5.测试原则不同 6](#_Toc10261)

[6.测试用例设计 6](#_Toc10237)

[五.测试报告 8](#_Toc20520)

[1.安装 8](#_Toc25323)

[2.程序代码检查 8](#_Toc16390)

[3.画面及报表格式检查 8](#_Toc762)

[4.功能测试 8](#_Toc762)

[5.正确性测试 8](#_Toc762)

[6.可靠性测试 8](#_Toc762)

一.测试方案的目的

  简单地说，就是替用户受过，测试的最终目的是确保最终交给用户的产品的功能符合用户的需求，把尽可能多的问题在产品交给用户之前发现并改正。具体地讲，测试一般要达到下列目标：

  1、确保产品是健壮的和适应用户环境的健壮性即稳定性，是产品质量的基本要求，尤其对于一个用于事务关键或时间关键的工作环境中。另外就是不能假设用户的环境(某些项目可能除外)，如：报业用户许多配置是比较低的，而且是和某些第三方产品同时使用的。

  2、确保产品满足性能和效率的要求使用起来系统运行效率低(性能低)、或用户界面不友好、用户操作不方便(效率低)的产品不能说是一个有竞争力的产品。用户最关心的不是你的技术有多先进、功能有多强大，而是他能从这些技术、这些功能中得到多少好处。也就是说，用户关心的是他能从中取出多少，而不是你已经放进去多少。

  3、确保产品完成了它所承诺或公布的功能，并且所有用户可以访问到的功能都有明确的书面说明------在某种意义上与ISO9001是同一种思想。产品缺少明确的书面文档，是厂商一种短期行为的表现，也是一种不负责任的表现。所谓短期行为，是指缺少明确的书面文档既不利于产品最后的顺利交付，容易与用户发生矛盾，影响厂商的声誉和将来与用户的合作关系;同时也不利于产品的后期维护，也使厂商支出超额的用户培训和技术支持费用。从长期利益看，这是很不划算的。领测认为接触过的软件产品，很少有向方正这样大大的产品、薄薄的文档。

  软件测试的目的是为了能增强软件使用可靠性，发现软件存在的不足和差异等等。要成为一名专业的软件测试人才必须要掌握专业技能因此要不断学习。

二、测试方案与计划的区别

1.定义不同：

测试计划是对测试过程的组织、资源、原则等进行规定和约束；而测试方案规则是描述所测软件的测试特性、测试方法、测试用例设计、测试代码设计、测试环境规划以及测试工具设计和选择的一种策略与方法。

2.层次不同：

测试计划是管理层面的，从组织管理的角度规划测试活动，而测试方案是技术层面的，从技术的角度规划测试活动。

总而言之，测试方案需要在测试计划的指导下进行，测试计划提出“做什么”，而测试方案明确“怎么做”。

1. 测试方案

1.测试需求分析

测试需求分析就是把产品需求和对用户的理解（用户体验）转化、分解成测试功能 点，产品需求是我们测试需求主要输入，但不是全部，我们还需要仔细分析产品设计说明，可以产出更多可测试的功能点（这些功能点往往没有包含在产品需求 中）。还要加入对性能、安全、接口和回归测试范围分析。测试需求是确定测试进度计划和资源的主要依据。评估风险并确定测试优先级。

## 2.测试内容

一个好的测试策略应该包括下列内容：

* 实施的测试类型和测试的目标
* 实施测试的阶段
* 技术
* 用于评估测试结果和测试是否完成的评测和标准
* 对测试策略所述的测试工作存在影响的特殊事项

## 3.测试进度计划

根据测试需求和策略，结合项目优先级和测试资源情况，评估测试进度计划，一般情况下，测试资源越充分，测试进度越乐观，但并非绝对，有时候一些软件ＢＵＧ会阻塞测试进度，这也是项目风险的一部分。

## 4.风险管理

在测试执行开始之前，对可能的风险进行分析和识别很重要，可以提前进行预防和采取应对措施，所以项目过程中，我们需要定期评估测试进度情况，提前进行风险预警。

## 5.质量

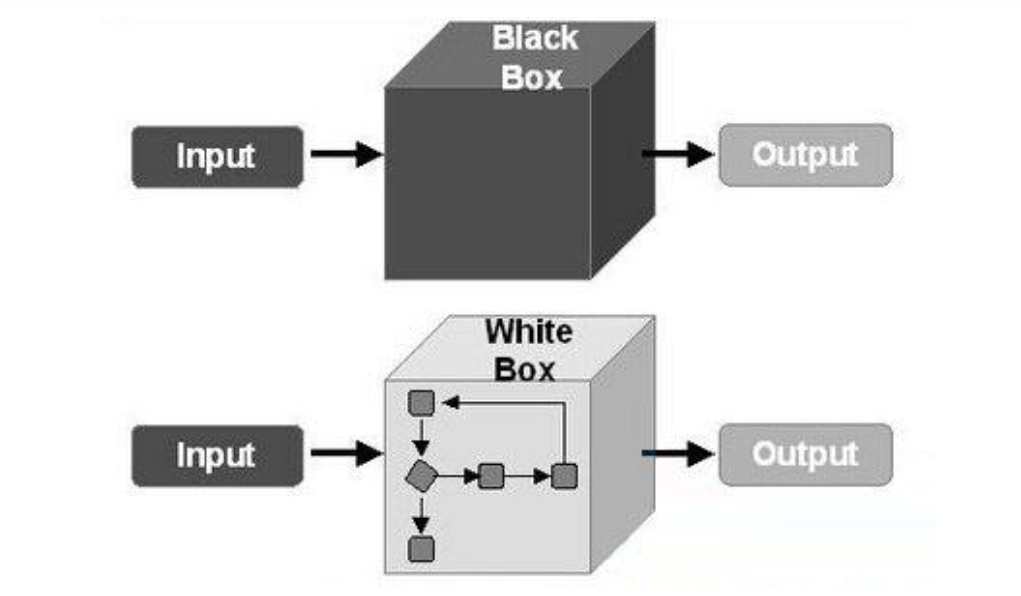
质量是指测试项目需要达到的标准，各个公司和项目都会有相应的标准要求，由于质量标准可以是公司内多数项目共识，所以也可以不必在测试方案中列出。对测试项目来说，比较常见的是以测试用例执行率、通过率和未关闭BUG级别/数量来设定质量标准。

四、测试策略

1、测试类型

软件测试分为功能测试和结构测试，即所说的黑盒测试和白盒测试。白盒测试是穷举路径测试，黑盒测试是穷举输入测试，这两种方法是基于完全不同的观点，反应了事物的两个极端，它们各有侧重和优势，但不能彼此替代。在现代的测试理念中，这两种测试方法不是截然分开的，而是交叉使用。

2、两者的比较



黑盒测试和白盒测试

从上图可以简要看出白盒测试和黑河测试的异同。白盒测试：白盒测试也称为结构测试，主要用于检测软件编码过程中的错误。程序员的编程经验、对编程软件的掌握程度、工作状态等因素都会影响到编程质量，导致代码错误。软件程序基本的语法错误在程序调试时，就能够很及时地发现，然后及时进行改正。但是软件程序在运算顺序、逻辑判断以及运行路径上的错误很难发现，在实际的软件程序代码编写上，没有程序员能够保证代码编写结构不出现任何错误，即使是水平很高的程序员也不能保证。白盒测试下，软件程序被看做是一个打开的盒子，盒子里有测试软件的源程序，还能够分析盒子内部的结构，所以这种测试方法能够全面地测试程序代码结构。黑盒测试：黑盒测试又称为功能测试，主要检测软件的每一个功能是否能够正常使用。在测试过程中，将程序看成不能打开的黑盒子，不考虑程序内部结构和特性的基础上通过程序接口进行测试，检查程序功能是否按照设计需求以及说明书的规定能够正常打开使用。

3.测试方式不同

黑盒测试：功能测试，是通过测试来检测每个功能是否都能正常使用。

(一)等价类划分法

等价类划分法是一种典型的、重要的黑盒测试方法，它将程序所有可能的输入数据划分为若干个等价类。然后从每个部分中选取具有代表性的数据当做测试用例。测试用例由有效等价类和无效等价类的代表数据组成，从而保证测试用例具有完整性和代表性。使用该方法设计测试用例主要有两个步骤：(1)确定等价类；(2)生成测试用例。

(二)边界值分析法

边界值分析法是对程序输入或输出的边界值进行测试的一种黑盒测试方法。实际的测试工作证明，考虑了边界条件的测试用例比那些没有考虑边界条件的测试用例具有更高的测试回报率。这里所说的边界条件，是指输入和输入等价类中那些恰好处于边界、或超过边界、或在边界以下的状态。

(三)因果图法

因果图法也是较常用的一种黑盒测试方法，是一种简化了的逻辑图。因果图能直观地表明输入条件和输出动作之间的因果关系，能帮助测试人员把注意力集中到与程序功能有关的输入组合上。因果图法是一种适合于描述对于多种输入条件组合的测试方法，根据输入条件的组合、约束关系和输出条件的因果关系，分析输入条件的各种组合情况，从而设计测试用例的方法，它适合于检查程序输入条件的各种组合情况

(四)错误推测法

错误推测法是基于以往的经验和直觉，参照以往的软件系统出现的错误，推测当前被测程序中可能存在的缺陷和错误，有针对性地设计测试用例。用错误推测法设计测试用例的基本思想是：列举出程序中可能犯出现的错误或容易发生错误的特殊情况的清单，然后根据清单和已经设计好的测试用例来编写特定的测试用例。例如，程序中出现的输入数据为“0”或者字符为空就是一种错误易发情况；在出现输入或输出的数量不定的地方，数量为“没有”和“一个”也是错误易发情况。特别需要注意的是，在阅读规格说明时联系程序员可能做的假设来确定测试用例，测试人员要站在用户的角度来考虑输入信息，而不必去管这些信息对于被测程序是合理还是不合理的输入。

白盒测试：称结构测试、透明盒测试、逻辑驱动测试或基于代码的测试。

1.是程序结构分析，根据源代码可以首先绘制程序的流程图，然后根据流程图分析程序的结构。

2.是逻辑覆盖方测试，根据程序的内部结构，对所有的路径进行测试，是一种穷举路径的测试方法。

3.是基本路径测试，根据程序的逻辑判断，分析程序中的路径，再进行用例的设计。

白盒测试是软件测试中比较重要的一种测试方法，可以分为四个步骤实施，第一步，撰写测试计划。根据需求说明书，制定软件测试的进度，确定人员、范围、技术、风险等，形成测试计划或测试方案。第二步，撰写测试用例。根据源代码及其分析，按照一定规范化的方法进行软件结构划分，并进行测试用例的设计，形成测试用例表。第三步，执行测试用例。按照之前写好的测试用例，进行系统测试的执行，并且记录测试结果，形成缺陷表和缺陷报告。第四步，撰写测试总结。将前期的测试工作做总结，分析用例的数量、发现的高、中、低缺陷数，评价本系统，形成完整的总结报告。

## 4.测试目的不同

1、黑盒测试：把程序看作一个不能打开的黑盒子，在完全不考虑程序内部结构和内部特性的情况下，在程序接口进行测试，只检查程序功能是否按照需求规格说明书的规定正常使用，程序是否能适当地接收输入数据而产生正确的输出信息。黑盒测试与白盒测试相反，软件设计程序被看做一个打不开的盒子，盒子里的程序代码测试人员不能看到，只能看到软件或者是某些模块的简单功能描述，这种测试方法主要是验证软件或者是功能的实现度。

2、白盒测试：通过检查软件内部的逻辑结构，对软件中的逻辑路径进行覆盖测试。在程序不同地方设立检查点，检查程序的状态，以确定实际运行状态与预期状态是否一致。

## 5.测试原则不同

1、黑盒测试：以用户的角度，从输入数据与输出数据的对应关系出发进行测试的。很明显，如果外部特性本身设计有问题或规格说明的规定有误，用黑盒测试方法是发现不了的。

2、白盒测试：一个模块中的所有独立路径至少被测试一次。所有逻辑值均需测试true和false两种情况。

## 6.测试用例设计

白盒测试用例设计

白盒测试关注的是测试用例执行的程度或覆盖程序逻辑结构（源代码）的程度。完全的白盒测试是将程序中每条路径都执行到，然而对一个带有循环的程序来说，完全的路径测试并不切合实际。白盒测试的特点：依据软件设计说明书进行测试、对程序内部细节的严密检验、针对特定条件设计测试用例、对软件的逻辑路径进行覆盖测试。

语句覆盖是最起码的结构覆盖要求，语句覆盖要求设计足够多的测试用例，使得程序中每条语句至少被执行一次。可以很直观地从源代码得到测试用例，无须细分每条判定表达式。由于这种测试方法仅仅针对程序逻辑中显式存在的语句，但对于隐藏的条件和可能到达的隐式逻辑分支，是无法测试的。（遗漏隐藏的逻辑分支判定覆盖要求必须编写足够的测试用例，使得每一个判断都至少有一个为“真”和为“假”的输出结果。判定覆盖比语句覆盖要多几乎一倍的测试路径，当然也就具有比语句覆盖更强的测试能力。同样判定覆盖也具有和语句覆盖一样的简单性，无须细分每个判定就可以得到测试用例。往往大部分的判定语句是由多个逻辑条件组合而成（如，判定语句中包含AND、OR、CASE），若仅仅判断其整个最终结果，而忽略每个条件的取值情况，必然会遗漏部分测试路径。（遗漏组合判定中的某些条件取值）条件覆盖要求设计足够多的测试用例，使得判定中的每个条件获得各种可能的结果，即每个条件至少有一次为真值，有一次为假值。要达到条件覆盖，需要足够多的测试用例，但条件覆盖并不能保证判定覆盖。条件覆盖只能保证每个条件至少有一次为真，而不考虑所有的判定结果。判定/条件覆盖要求设计足够多的测试用例，使得判定中每个条件的所有可能结果至少出现一次，每个判定本身所有可能结果也至少出现一次。判定/条件覆盖满足判定覆盖准则和条件覆盖准则，弥补了二者的不足。判定/条件覆盖准则的缺点是未考虑条件的组合情况。多重条件覆盖要求设计足够多的测试用例，使得每个判定中条件结果的所有可能组合至少出现一次。多重条件覆盖准则满足判定覆盖、条件覆盖和判定/条件覆盖准则。更改的判定/条件覆盖要求设计足够多的测试用例，使得判定中每个条件的所有可能结果至少出现一次，每个判定本身的所有可能结果也至少出现一次。并且每个条件都显示能单独影响判定结果。缺点是线性地增加了测试用例的数量。路径覆盖要求设计足够的测试用例，覆盖程序中所有可能的路径。由于路径覆盖需要对所有可能的路径进行测试（包括循环、条件组合、分支选择等），那么需要设计大量、复杂的测试用例，使得工作量呈指数级增长。而在有些情况下，一些执行路径是不可能被执行的，这样不仅降低了测试效率，而且大量的测试结果的累积，也为排错带来麻烦。

黑盒测试用例设计

等价类划分是把所有可能的输入数据,即程序的输入域划分成若干部分（子集）,然后从每一个子集中选取少数具有代表性的数据作为测试用例。等价类分为有效等价类和无效等价类，等价类划分有以下原则：在输入条件规定了取值范围或值的个数的情况下,则可以确立一个有效等价类和两个无效等价类。在输入条件规定了输入值的集合或者规定了"必须如何"的条件的情况下，可确立一个有效等价类和一个无效等价类。在输入条件是一个布尔量的情况下,可确定一个有效等价类和一个无效等价类。在规定了输入数据的一组值（假定n个）,并且程序要对每一个输入值分别处理的情况下,可确立n个有效等价类和一个无效等价类。例：输入条件说明学历可为:专科、本科、硕士、博士四种之一，则分别取这四种这四个值作为四个有效等价类，另外把四种学历之外的任何学历作为无效等价类。在规定了输入数据必须遵守的规则的情况下,可确立一个有效等价类（符合规则）和若干个无效等价类（从不同角度违反规则）；在确知已划分的等价类中各元素在程序处理中的方式不同的情况下,则应再将该等价类进一步的划分为更小的等价类。在确立了等价类后,可建立等价类表,列出所有划分出的等价类输入条件：有效等价类、无效等价类，然后从划分出的等价类中按以下三个原则设计测试用例：为每一个等价类规定一个唯一的编号；设计一个新的测试用例,使其尽可能多地覆盖尚未被覆盖地有效等价类,重复这一步，直到所有的有效等价类都被覆盖为止；设计一个新的测试用例,使其仅覆盖一个尚未被覆盖的无效等价类,重复这一步，直到所有的无效等价类都被覆盖为止。

边界值分析法就是对输入或输出的边界值进行测试的一种黑盒测试方法。通常边界值分析法是作为对等价类划分法的补充，这种情况下，其测试用例来自等价类的边界。边界值分析不是从某等价类中随便挑一个作为代表，而是使这个等价类的每个边界都要作为测试条件。边界值分析不仅考虑输入条件，还要考虑输出空间产生的测试情况。

因果图是一种利用图解法分析输入的各种组合情况，从而设计测试用例的方法，它适合于检查程序输入条件的各种组合情况。场景法通过流程图描述用户的使用场景；适用场景:多个功能组合场景的测试

判定表适用场景：有多个输入与输出，并且输入与输入之间有组合关系，输入与输出之间有依赖关系；组成部分：条件桩（所有的输入条件，次序无关）、动作桩（输入条件对应可能的结果）；条件项（条件对应的取值，一般是真假值）、动作项（取值全组合，每一列组合得到的结果）；规则（每列条件项和动作项构成一条规则）；设计用例步骤：明确需求、画出判定表、根据规则编写用例。

五.测试报告

项目名称:分类8新闻分类系统

**1.安装**

(1）程序运行环境已经正确设定。 是

**2.程序代码检查**

(1）程序单位首部有程序说明和修改备注。 是

(2）变量、过程、函数命令符合规则。 是

(3）程序中有足够的说明信息。 是

(4)修改注释符合要求。 是

(5）类库的使用符合要求。 是

**3.画面及报表格式检查**

(1）画面和报表格式符合规定需求。 是

(2）程序命名符合格式需求。 是

(3）画面和报表的字段位置和宽度与设计文档一致。 是

**4.功能测试**

(1）多画面之间切换正确。 是

(2）功能键、触发键、按钮、菜单、选择项功能正确。 是

(3）数据项关联及限制功能正确。 是

**5.正确性测试**

(1）读/写/删除操作结果正确。 是

(2）各种组合条件之查询或报表正确。 是

(3）设计文档规定的其它操作。 是

**6.可靠性测试**

(1）非法键容错测试。 是

(2）异常字符容错测试。 是

(3）程序负作用检查。 是