**中图分类号：TP393**

**论文编号：10006SY1306314**



硕 士 学 位 论 文

**适航领域软件需求跟踪方法研究与应用**

作者姓名 赵腾

学科专业 计算机技术

指导教师 曹庆华 教授

培养院系 计算机学院

**Study and Implementation of the Network Anomaly Detection Techniques Based on Entropy**

A Dissertation Submitted for the Degree of Master

**Candidate：Qin Xi**

**Supervisor：Assoc. Prof. Xu Tongge**

School of Computer Science & Engineering

Beihang University, Beijing, China

**中图分类号：TP393**

**论文编号：10006SY1306314**

硕 士 学 位 论 文

基于信息熵的网络异常检测技术

研究与实现

作者姓名 秦希 申请学位级别 工学硕士

指导教师姓名 徐同阁 职 称 副教授

学科专业 计算机应用技术 研究方向 网络管理

学习时间自 2013年 9月 5日 起至 2015年 12月 19日止

论文提交日期 2015年 12月 22日 论文答辩日期 2015年 12月 19日

学位授予单位 北京航空航天大学 学位授予日期 2016年 1月 8 日

关于学位论文的独创性声明

本人郑重声明：所呈交的论文是本人在指导教师指导下独立进行研究工作所取得的成果，论文中有关资料和数据是实事求是的。尽我所知，除文中已经加以标注和致谢外，本论文不包含其他人已经发表或撰写的研究成果，也不包含本人或他人为获得北京航空航天大学或其它教育机构的学位或学历证书而使用过的材料。与我一同工作的同志对研究所做的任何贡献均已在论文中做出了明确的说明。

若有不实之处，本人愿意承担相关法律责任。

学位论文作者签名： 日期： 年 月 日

学位论文使用授权书

本人完全同意北京航空航天大学有权使用本学位论文（包括但不限于其印刷版和电子版），使用方式包括但不限于：保留学位论文，按规定向国家有关部门（机构）送交学位论文，以学术交流为目的赠送和交换学位论文，允许学位论文被查阅、借阅和复印，将学位论文的全部或部分内容编入有关数据库进行检索，采用影印、缩印或其他复制手段保存学位论文。

保密学位论文在解密后的使用授权同上。

学位论文作者签名： 日期： 年 月 日

指导教师签名： 日期： 年 月 日

**摘 要**

关键词：，，，

**Abstract**

**Key words:**, ,

**目 录**

[第一章 绪论 1](#_Toc438730721)

[1.1 研究背景及意义 1](#_Toc438730722)

[1.2 研究现状 2](#_Toc438730723)

[1.3 论文研究内容 4](#_Toc438730724)

[1.4 论文组织结构 4](#_Toc438730725)

[第二章 网络异常检测技术 6](#_Toc438730726)

[2.1 网络异常的概念与分类 6](#_Toc438730727)

[2.1.1 DDoS攻击 7](#_Toc438730728)

[2.1.2 端口扫描 8](#_Toc438730729)

[2.1.3 蠕虫病毒 8](#_Toc438730730)

[2.1.4 其他常见攻击 9](#_Toc438730731)

[2.2 网络异常检测技术 10](#_Toc438730732)

[2.2.1 基于统计学的检测方法 10](#_Toc438730733)

[2.2.2 基于机器学习的检测方法 11](#_Toc438730734)

[2.2.3 基于信息熵的检测方法 12](#_Toc438730735)

[2.2.4 当前异常检测技术的不足 13](#_Toc438730736)

[2.3 本章小结 14](#_Toc438730737)

[第三章 基于多特征信息熵预测的异常检测算法 15](#_Toc438730738)

[3.1 算法特征选择 15](#_Toc438730739)

[3.1.1 Flow格式 15](#_Toc438730740)

[3.1.2 信息熵 16](#_Toc438730741)

[3.1.3 特征选择 17](#_Toc438730742)

[3.2 基于多维信息熵预测的异常检测 21](#_Toc438730743)

[3.2.1 预处理阶段 22](#_Toc438730744)

[3.2.2 检测阶段 23](#_Toc438730745)

[3.3 实验与分析 26](#_Toc438730746)

[3.3.1 实验数据 26](#_Toc438730747)

[3.3.2 实验结果与分析 27](#_Toc438730748)

[3.3.3 实验总结 32](#_Toc438730749)

[3.4 本章小结 33](#_Toc438730750)

[第四章 基于信息熵聚类的异常检测算法 34](#_Toc438730751)

[4.1 基于信息熵聚类的异常检测 34](#_Toc438730752)

[4.1.1 检测原理 34](#_Toc438730753)

[4.1.2 聚类模型构建 35](#_Toc438730754)

[4.1.3 异常识别策略 36](#_Toc438730755)

[4.2 实验及分析 37](#_Toc438730756)

[4.2.1 实验数据 37](#_Toc438730757)

[4.2.2 实验结果与分析 38](#_Toc438730758)

[4.2.3 实验总结 41](#_Toc438730759)

[4.3 本章小结 42](#_Toc438730760)

[第五章 网络异常检测系统的设计与实现 43](#_Toc438730761)

[5.1 系统设计目标 43](#_Toc438730762)

[5.2 开发环境 43](#_Toc438730763)

[5.3 系统总体设计 44](#_Toc438730764)

[5.3.1 系统架构 44](#_Toc438730765)

[5.3.2 系统组件设计 46](#_Toc438730766)

[5.3.3 数据库设计 47](#_Toc438730767)

[5.4 系统设计与实现 48](#_Toc438730768)

[5.4.1 流量拆分 48](#_Toc438730769)

[5.4.2 信息熵计算 50](#_Toc438730770)

[5.4.3 模型构建 50](#_Toc438730771)

[5.4.4 异常检测 52](#_Toc438730772)

[5.4.5 流量分析 52](#_Toc438730773)

[5.5 本章小结 54](#_Toc438730774)

[第六章 系统验证及结果分析 55](#_Toc438730775)

[6.1 实验介绍 55](#_Toc438730776)

[6.1.1 部署环境 55](#_Toc438730777)

[6.1.2 实验数据 55](#_Toc438730778)

[6.2 系统功能验证 56](#_Toc438730779)

[6.2.1 异常检测过程 56](#_Toc438730780)

[6.2.2 分析异常流量 56](#_Toc438730781)

[6.3 本章小结 60](#_Toc438730782)

[结 论 61](#_Toc438730783)

[参考文献 63](#_Toc438730784)

[攻读硕士学位期间取得的学术成果 66](#_Toc438730785)

[致 谢 67](#_Toc438730786)

图 目 录

[图 1 DDoS攻击体系结构 7](#_Toc438740123)

[图 2 异常引起的目的IP地址和目的端口的分布变化 19](#_Toc438740124)

[图 3 异常引起的信息熵序列变化情况 20](#_Toc438740125)

[图 4 基于多特征信息熵预测的异常检测算法流程框架 22](#_Toc438740126)

[图 5 阈值的设定原理 25](#_Toc438740127)

[图 6 两处新检测出的DDoS对应的相关特征曲线 28](#_Toc438740128)

[图 7 异常占比对应检测率情况 30](#_Toc438740129)

[图 8 误报前后流量及总偏移量变化情况 31](#_Toc438740130)

[图 9 误报时间单元包大小分布 31](#_Toc438740131)

[图 10 基于信息熵聚类的异常检测算法框架图 34](#_Toc438740132)

[图 11 判断异常发生流程图 36](#_Toc438740133)

[图 12 聚类数目k与类簇指标的变化关系 38](#_Toc438740134)

[图 13 基于信息熵聚类算法与Improved-ETS算法的结果对比 40](#_Toc438740135)

[图 14 训练集容量与检测率对应图 41](#_Toc438740136)

[图 15 网络流量采集与异常检测整体架构图 45](#_Toc438740137)

[图 16 系统组件图 46](#_Toc438740138)

[图 17 Flow数据预处理组件 47](#_Toc438740139)

[图 18 网络异常检测系统主要数据实体关系图 48](#_Toc438740140)

[图 19 模型构建实现流程图 51](#_Toc438740141)

[图 20 模型构建相关类图 51](#_Toc438740142)

[图 21 流量分析路径 53](#_Toc438740143)

[图 22 TopN通信对象分析的SQL语句设计 54](#_Toc438740144)

[图 23 模型管理、异常检测与流量分析模块包图 54](#_Toc438740145)

[图 24 实验部署环境 55](#_Toc438740146)

[图 25 异常检测结果 56](#_Toc438740147)

[图 26 流量曲线图 57](#_Toc438740148)

[图 27 DDoS异常时段TopN协议分析 58](#_Toc438740149)

[图 28 DDoS异常时段TopN用户分析 58](#_Toc438740150)

[图 29 DDoS异常时段疑似被攻击用户TopN通信对象分析 59](#_Toc438740151)

[图 30 DDoS异常时段疑似被攻击用户TopN端口分析 59](#_Toc438740152)

[图 31 DDoS异常时段疑似被攻击用户半连接分析 59](#_Toc438740153)

[图 32 DDoS异常时段疑似被攻击用户包大小分布分析 60](#_Toc438740154)

表 目 录

[表 1 几种典型的蠕虫特征 9](#_Toc438740155)

[表 2 Flow主要字段表 16](#_Toc438740156)

[表 3 Flow源字段与所选算法特征对应情况 21](#_Toc438740157)

[表 4 时间单元*T*i所占比例*Pi*时主要特征拆分变化策略 22](#_Toc438740158)

[表 5 异常类型判别原理 25](#_Toc438740159)

[表 6 DARPA中部分DDoS攻击举例 27](#_Toc438740160)

[表 7 本章Improved-ETS与改进前ETS算法针对DARPA的检测结果对比 28](#_Toc438740161)

[表 8 R2L攻击造成的偏移量分析 29](#_Toc438740162)

[表 9 误报时间单元偏移量分析 31](#_Toc438740163)

[表 10 聚类结果在熵空间的分布情况 39](#_Toc438740164)

[表 11 训练样本容量与最佳聚类数目对应表 41](#_Toc438740165)

[表 12 系统开发平台与主要工具 44](#_Toc438740166)

[表 13 时间单元*T*i所占比例*Pi*时各个特征拆分变化策略 49](#_Toc438740167)

[表 14 模型训练基本信息 50](#_Toc438740168)

[表 15 模型构建重要方法或属性说明 52](#_Toc438740169)

[表 16 正常流量过滤结果 56](#_Toc438740170)

# 绪论

本章首先阐述了论文的研究背景以及研究意义；其次对目前国内外关于适航领域软件需求跟踪所使用的方法进行了总结和分析，对比了各种方法的优劣以及适用场景，指出了目前存在的问题以及未来的发展趋势；之后针对这些问题明确了论文的研究目标和研究内容，最后介绍了整篇论文的组织结构。

## 课题来源及研究背景

本课题来源于工信部民机专项——64 位高安全综合化航空机载计算机项目子课题FADEC适航关键技术研究（YWF-15-GJSYS-087）。

2017年商飞C919大型客机的试飞成功，标志着我国的民用航空事业跨入了新的历史阶段。民用航空事业的基本理念是“民机发展，适航先行”。所谓“适航（适航性）”是指航空器包括部件和子系统的整体性能和操纵特性能够在预期的环境下安全飞行（包括起飞和着陆）的固有品质，并且这种品质能够通过航空器全寿命周期的设计、制造、试验、使用、为何和管理等环节来实现和保持。适航性即安全性，是民机能够投入使用、走向市场的重要前提和保证。

随着计算机技术的发展以及在航空领域的应用，机载软件承担了越来越多的功能，同时软件的种类和规模也持续增长。*机载软件规模增长和大量使用主要有三个原因：1）通过更加智能和方便的软件来改善人机界面最终减轻飞行员的负担；2）软件设备的重量、功耗、维护成本往往低于其他类型的设备；3）软件和硬件的功能分配比例向软件倾斜。*机载软件在机载系统乃至整机中起到了越来越重的作用，但是由于规模的增长，增大了软件的研制和审定的工作量，如何保证软件在整个生命周期过程中的需求可跟踪性成为影响项目成败的重要因素。2011年航空无线电技术委员会（RTCA）在DO-178B的基础上提出了DO-178C，即机载系统和设备合格审定中的软件考虑。该标准以可协调的方法和可接受的置信度，为航空界确定机载系统和设备中的软件是否符合适航要求提供了指南，在标准层面上约束了软件开发的流程以及应该达到的目标，其中应该达到的目标中明确指出：1）高层需求可追踪到系统需求；2）底层需求可追踪到高层需求；3）源代码可追踪至底层需求。

需求跟踪，即通过建立需求同系统元素的之间的联系，跟踪在整个软件生命周期中某一需求的使用全过程。在工业实践中，现有的需求跟踪技术仍然是通过填写需求跟踪矩阵和需求跟踪图的方式来进行跟踪，这种方式比较精确，但是工作量也随着软件规模的增长而加大，同时在变更需求和增加需求的时候，难以扩展且容易引入错误。市面上的需求管理软件有：IBM Rational DOORS、IBM Rational RequisitePro和青铜器RDM等，这些软件能够帮助软件相关人员改进项目目标的沟通，增强协作开发。同样的，这些软件的需求跟踪也是通过建立需求跟踪矩阵来实现，尽管软件能够给信息录入人员简化一部分操作，但当软件规模逐渐增大时，工作量也会持续增长。因此，能够自动化或者半自动化地帮助软件开发人员简化相关工作具有非常重要的意义。在学术领域，需求跟踪自动化技术主要使用信息检索的方式，使用向量空间模型（VSM）、隐形语义索引（LSI）和概率模型（PM）等模型[2]。这些方法基本是基于统计学理论对软件文档进行归类分析，很少考虑软件文档的语义，同时已有的方法只适用于普通软件，并不能对特定领域（如适航领域、医学领域等）的软件文档进行分析。近年来越来越热门的自然语言处理（NLP）能够在基于统计计算的同时，加入语义关系的计算，从而提升文档关系计算的准确率。因此，在软件需求跟踪的过程中加入语义计算，一定能够提升需求跟踪关系的精确程度。

综上所述，软件需求跟踪对于适航领域的适航性目标的达成具有重要的意义，同时对自动化的需求跟踪技术的研究更是对规模日益增长的软件系统研制和审定具有重要意义。

## 国内外系统需求

### 需求跟踪技术概述

需求跟踪技术分为两类：静态需求跟踪技术、动态需求跟踪技术。静态需求跟踪技术主要使用跟踪矩阵、跟踪图和交叉引用等多种形式表示跟踪过程，静态需求跟踪链静态表示；动态需求跟踪技术是在软件开发过程和需求变更时自动建立或维护跟踪关系，主要使用基于信息检索、基于事件触发、基于规则的方法[4]。

1. 静态需求跟踪技术

传统需求跟踪主要使用静态跟踪技术，但随着软件规模越来越大、软件声明周期越来越长，静态跟踪技术易于出错、不易维护的缺点暴露出来，如果继续使用则成为软件维护的一个负担。

需求跟踪矩阵（Requirement Traceability Matrix， RTM）用来表示需求和其他系统元素的联系链，每个功能性需求向后连接一个特定的使用实例，向前连接一个或多个设计、代码和测试元素。设计元素可以是模型中的对象，例如数据流图、关系数据模型中的表单、或对象类。代码参考可以是类中的方法，源代码文件名、过程或函数等。跟踪能力联系链可以定义各种系统元素类型间的一对一，一对多，多对多关系。需求跟踪矩阵示例如表所示。

表 需求跟踪矩阵示例



需求跟踪图可以直观的表示需求之间的联系和需求和各个阶段元素之间的关系，以及在需求变更时对各个部分的影响。不仅支持软件需求到系统设计文档、系统相关的规格说明、实现代码之间的跟踪，还支持软件开发过程中的中间制品之间的跟踪。在跟踪图中，用户可以自己定义图中的对象和关系，方便用户使用自己熟悉的语言对关系描述。

交叉引用主要应用在需求文档之间建立跟踪关系，可以在系统的需求说明文档、软件需求规格说明文档等中间建立需求跟踪关系。此方法建立的跟踪方法较直观，便于实际使用，但局限性是只适用于需求文档之间跟踪关系的建立[5]。

1. 动态需求跟踪技术

动态需求跟踪针对手工建立需求跟踪代价过高且易于出错的问题,以自动化技术为手段辅助开发人员建立和维护跟踪关系[4]。

软件生命周期内出现的大部分软件制品都包含着大量格式化与非格式化的文本信息因此主流的动态需求跟踪方法大多将软件制品视为待检索的文档，并计算不同软件制品之间的相似度,对于相似度高于一定阈值的软件制品则之间存在跟踪关系[6]。

[4]中介绍了一种基于句法分析的跟踪关系恢复方法，通过句法分析我们可以识别出最有可能刻画软件制品特征的部分动词与名词，并减少制品中存在的噪音对需求跟踪关系恢复过程所造成的不利影响。方法分为如下步骤：1）将制品中的句子切分为句子块（代码特殊处理）；2）词性标注；3）对句子进行块分析，利用句子的上下文来修正词性标注过程中可能引入的错误；4）减少标引词中存在的噪声；5）通过聚类簇映射为分类建立映射关系。该方法关键是使用语义聚类对得到的不同分类进行映射，然后通过得到的结果来建立跟踪关系。

[7]和[4]是同一个作者，[7]的是从命名实体识别的角度来恢复需求追踪关系，主要解决需求和源代码的跟踪关系。该文章提出基于文本的需求追踪方法严重依赖于文本质量[7]，因此，文中提出一种方法，通过代码实体上下文构建命名实体识别模型，解决了抽象语法树和正则表达式无法解析非源代码形式的软件制品问题。文中使用基于最大熵模型的需求追踪方法，将源代码中的命名实体识别看做分类问题，同时代码的类别包括：class、method、invocate、comment、param、normalText，建立分类模型后对代码进行分类；然后对上述已经找到命名实体的制品文本进行预处理，包括分词、过滤停用词和词干化处理，将软件制品转为文档集合，然后使用[4]中相同的聚类方法对标引词进行分类和映射，从而建立跟踪关系。

[6]提出使用代码注释辅助动态需求跟踪的方法，用来建立需求文档和代码之间的跟踪关系。文中指出动态需求跟踪运用信息检索等技术，自动化建立需求文档和工作产品的跟踪关系，在跟踪精度等方面仍然存在问题，文中认为问题一部分是出在没有利用注释信息上面。使用的方法和[7]相似，主要有两点不同：1）预处理数据阶段，使用自动翻译工具将中文的软件需求文档翻译成英文，因为文中认为代码都是英文，也省去了中文分词的步骤；2）使用向量空间模型对文本进行检索，得到文本和代码的相似度，达到一定阈值的即可建立跟踪关系。

[8]提出基于LSI（Latent Semantic Indexing），即潜在语义索引，重建需求和设计制品、测试用例之间的跟踪关系。LSI是信息检索技术中的一种，可以将信息降维，文中提出一旦所有的文档都表示为LSI的子空间，那么就能通过余弦相似度方法计算文档间的关系。同时指出LSI不依赖于提前定义的单词表或语法，也就是说，可以省去花费巨大的数据预处理过程。

[9]是比较早提出通过信息检索的方式来增强需求跟踪的文章。文中主要工作包括：1）将问题构造成信息检索问题；2）选择IR算法；3）准备算法输入的需求文本；4）分析算法输出；5）选择合适的策略整理算法输出；6）比较算法性能和人工性能。比较重要的是，文中提出可能在这个过程中存在的问题：1）需求文档不全或语义模糊；2）有些缩略语没有定义；3）领域或工程知识缺少；4）高层需求和底层需求中使用不同的术语表示。这篇文章使用VSM算法，同时增添了两个扩展：一个是使用关键词列表，另一个使用简单的词典。

### 软件工程领域文本检索算法概述

在软件工程领域，越来越多的任务使用文本检索技术，比如：需求跟踪、特征定位、软件复用等。为了提升文本检索的性能，很多方法被提出，下面将列出一些与本文工作相关的一些方法。

首先是自动查询扩展技术（Automatic Query Expansion， AQE），该技术已经广泛地用在信息检索任务中。自动查询扩展技术通过对查询单词或短语的扩展，去解决“词汇问题”[14 The vocabulary problem in human-system communication]。所谓“词汇问题”指的是由于人类语言的多样性导致的查询语句中的单词与文档集中单词的不一致性问题，这也是“一词多义”或“一义多词”现象。C.Carpineto[13 A survey of automatic query expansion in information retrieval] 指出大部分的自动查询扩展都显示地利用词条的依赖特性，比如在建立同义词词典时使用词语的共现关系、语法关系等。除此之外，还有一些关于自动查询扩展的策略，比如对扩展词进行加权，使得不同的扩展词具有不同的重要程度，保证扩展后的内容更能完整的表示待扩展词的含义[15 A Study of Measure for Document Relatedness Evaluation]。

然后是词向量（本文中词向量特指Word Embedding 或 Word Vector），其被大量研究者用在词的表示和文档表示上。Word2vec是Google的Tomas Mikolov等人提出的文本表示方法，将单词表示成词向量的形式。训练词向量的模型有两种： CBOW和skip-gram，CBOW模型的输入是特定词上下文相关词的词向量，输出是该特定词的词向量；skip-gram模型与CBOW相反，输入是特定词的词向量，输出是特定词对应的上下文词的词向量。二者具体的实现过程将在第二章详细描述。

在软件工程很多文本检索任务都使用了word2vec或word2vec的改进模型。X. Ye等人[11 From Word Embeddings to Document Similarities for Improved Information Retrieval in Software Engineering]提出一种学习词向量的方式并且使用词向量去计算文档之间的相似度。在论文中，出于对一义多词现象的考虑，他们提出了两种不同的训练设定：一种是单词典设定，另一种是双词典设定。单词典设定是将自然语言文本和源代码混在一起，而双词典设定是将二者分开。举例说明：对于单词“clear”，在自然语言文本中是形容词，以为“干净的”或者动词“清除”等含义，在源代码文本中，该词为一个方法名。二者的不同是单词典设定会使用方法名“clear”在代码中的上下文去训练形容词“clear”，而双词典设定不会。然而，他们没有考虑很多软件文档是自然语言文本和源代码是混杂在一起的，因此他们的方法并不是对所有的软件制品适用。

Jin G.等人[6 Semantically enhanced software traceability using deep learning techniques.]通过对word2vec的改进去做跟踪链接的恢复，该方法将word2vec结合了循环神经网络（Recurrent Neural Networks，RNN），同时两个制品的语义相似度可以通过RNN预测。他们的方法在大规模的工业软件数据集上效果比较好，但是，对于一些相对较小的软件数据集，性能并没有预期的那么好。事实上，得到包含有足够训练的跟踪链接的大规模工业软件数据也是比较困难的。

另一个和[6]相似的工作是使用机器学习分类器去估计软件中跟踪链接的数量[3 Empirical Principles and an Industrial Case Study in Retrieving Equivalent Requirements via Natural Language Processing Techniques]。[3]中作者强调了使用分类器去对跟踪链接进行分类，而不是通过设置阈值来过滤跟踪链接的原因是自然语言处理方法在不同的数据集上表现各不相同，从而对于每一个软件数据集进行恢复跟踪链接时都有一个不同的参数。因此，在本文中我们也被启发结合使用机器学习方法和文本检索一起，去恢复软件需求跟踪链接。

Tien-Duy B. Le等人[35 Information retrieval and spectrum based bug localization: better together]在bug localization中使用信息检索和program spectrum，使用向量空间模型（Vector Space Model， VSM）、论文中提出的方法Tarantula和基于可疑词的方法计算三种不同的相似度，然后将三者通过三个参数结合起来。Shaowei Wang等人[36 AmaLgam+: Composing Rich Information Sources for Accurate Bug Localization]提出一个相似的方法去定位故障，从五个维度，即五种不同的源文件去定位故障，然后将五个结果整合在一起。这两种方法的共同点是从不同的角度去解决同一个问题，然后通过参数将不同的解法整合在一起，再通过随机梯度下降的方式对超参数进行求解。故障定位的任务和本文的需求跟踪任务相似，在实验部分将对比本文的方法和以上故障定位的方法。

另一个相关的工作是[41 An unsupervised approach for discovering relevant tutorial fragments for APIs]，该工作使用主题模型和PageRank算法去发现相关指导片段去完善API（Application Programming Interface,应用程序编程接口）。他们的方法使用非监督学习的方法，克服了监督学习中需要大量标注数据和对数据集高度的依赖等缺点，但是由于他们的方法使用了很多文档片段的特征，因此很难迁移到需求跟踪任务中。

除此之外，软件制品，包括需求文档、设计文档和源代码等，包含大量自然语言文本与代码的组合、代码片段和一些专有名词。这就导致了其与普通文章不同，对于普通文章适用的一些文本检索方法并不适用软件制品数据，或不能直接应用在软件制品数据上。

### 存在的问题

## 高级需求

本文的研究目标是研究机载软件生命周期数据跟踪关系的构建方法，通过建立模型，能够自动判断各生命周期的数据是否能够满足DO-178C中规定的可追溯性目标，为适航软件的审定提供部分依据。具体来说，包括对DO-178C中规定的可追溯性目标的研究和适航审定对该部分的具体要求、软件需求跟踪模型的研究与建立。本文的主要研究内容关系如图所示。



图 主要研究内容关系图

具体来说，主要研究内容包括：

（1）适航理论研究。DO-178C标准中对软件生命周期的定义与普通软件的有所不同，对系统和软件的可追溯性具有严格的要求。本文将对适航标准DO-178C及其补充标准进行研究，弄清楚在机载软件的软件生命周期，以及每个过程中应该达到的目标，重点在于标准中与需求跟踪有关的部分，在软件设计、开发到使用的全过程需要注意的问题。

（2）需求跟踪算法模型构建。本文在算法构建阶段重点关注使用基于词向量的文本语义相似度计算的有效性和学习排序在需求跟踪任务中的效果。算法构建过程主要包括：领域文档处理、文本语义相似度算法改进和利用学习排序算法提升需求跟踪算法性能。

（3）模型验证。从多个维度对提出的模型进行验证，将模型拆分成两个部分，分别对两部分的算法进行验证，并和当前一些领先的方法进行对比；最后将模型整体与国际领先的AML算法进行效果对比。实验使用开源的、经常用于跟踪恢复的文本数据集，采用F-measure、MRR和MAP等指标对模型进行评估。同时，实现了基于word embedding的适航领域软件需求跟踪原型系统。

综上所述，本文通过在需求跟踪任务中引入词向量、加权策略和查询扩展的方法对文本相似度算法进行了改进，并使用学习排序算法对需求跟踪链接进行进一步的优化，提出了一个面向适航领域软件文档的需求跟踪模型，并完成了模型的构建、分析、验证和展示。

## 低级需求

本文共分六章，各章节编排以及主要内容如图所示：



图 章节组织及各章节主要内容

第一章，绪论。首先给出了课题来源，介绍了论文研究背景及意义，阐述和总结了软件需求跟踪技术的国内外研究现状，同时还介绍了本文的研究目的和主要研究内容，最后简述了文章的组织结构。

第二章，相关理论与技术研究。本章围绕软件需求跟踪任务流程中的关键技术，调研了国内外需求跟踪任务的研究现状和同样使用信息检索技术解决的与需求跟踪任务相似的一些任务的国内外研究现状，以及软件工程领域对学习排序算法的应用和改进。通过对相关理论和技术的研究，确定了本文研究工作的立足点和一些理论依据。

第三章，适航领域软件需求跟踪算法模型构建。本章提出了一个新的基于word embedding和学习排序算法的软件需求跟踪算法模型Tr-WELR（Traceability using Word Eembedding and Learning to Rank）。首先提出了在软件需求跟踪任务中使用加权策略和查询扩展技术，然后使用word embedding改进了文本语义计算算法，最后使用学习排序算法对排序结果进行二次使用，提升需求跟踪任务的精确率。

第四章，适航领域软件需求跟踪算法模型验证。本章通过多组对比试验对提出的模型进行验证。首先对数据的收集流程和实验数据集做了介绍，然后详细描述了文档预处理的方法，接下来介绍了实验的一些设置，包括对比方法、词向量训练方法和实验环境，之后对模型的评估指标做了介绍，最后对实验结果和运行时间效率进行了分析和说明。

第五章，适航领域软件需求跟踪算法模型原型系统。本章介绍了原型系统的需求、开发环境和系统的总体设计，然后对原型系统中的主要功能模块进行介绍，最后对该原型系统功能进行展示。

总结与展望。总结本论文的研究工作，并针对不足之处提出今后的研究点和工作重点。

# 相关理论与技术研究

本章主要对适航领域相关理论、软件需求跟踪领域理论和相关技术进行描述。首先

## 源代码

本节首先介绍适航领域软件相关的标准，以及标准中规定的与普通软件开发不同的软件生命周期和相应的软件生命周期数据，然后介绍标准中与软件需求跟踪相关的内容。

### DO-178B/C标准介绍

20年代70年代末，计算机应用在飞机设备和系统中使用越来越多，因此为了保证这些应用能够满足适航局的审定要求，美国航空无线电技术委员会（Radio Technical Commission for Aeronautics，RTCA）成立特别委员会，专门研究制定软件规则，用以支持以软件为基础的设备和系统的合格审定[1]。自此之后，DO-178系列《机载系统和设备合格审定中的软件考虑》产生，并不断补充扩展，经历了DO-178、DO-178A、DO-178B，现在已经发展到DO-178C。其中DO-178B标准是相对稳定的，其制定汲取了多方面对共同意见，包括飞行制造商、设备供应商、工具开发商和适航认证机构等，在使用的19年过程中没有发现严重的问题。由于在实际开发软件的过程中，方法会因项目的不同而不同，因此DO-178B标准是面向过程和目标的，即尽可能的不涉及到实际的技术，但是，DO-178B在一些方面，“面向目标”的原则贯彻的不够彻底，因此对DO-178B做了一些修正，于是出现了DO-178C。

DO-178B/C规定的软件生命周期过程包括：软件计划过程、软件开发过程和软件综合过程。三个综合过程结构图如图所示。其中软件计划过程规定了五个计划和三个标准。五个计划包括：软件审定计划、软件开发计划、软件验证计划、软件构型管理计划、软件质量保证计划；三个标准包括：软件需求标准、软件设计标准和软件编码标准。软件开发过程包括：软件需求过程、软件设计过程、软件编码过程和软件集成过程。软件综合过程包括：软件验证过程、软件构型管理过程、软件质量保证过程和审定联络过程。

在整个软件生命周期中，软件开发过程是主线，软件综合过程与软件开发过程并行执行，且软件综合过程执行在软件开发过程的各个子过程上。具体的关系如方阵图图所示。虽然标准中写的软件生命周期过程较多，但是标准并没有强制所有软件活动必须按照规定的各活动的相互关系，只需要在软件生命周期中描述清楚所有过程和活动的先后顺序和执行关系，并定义过程之间的迁移准则即可。



图 DO-178B/C生命周期过程结构图

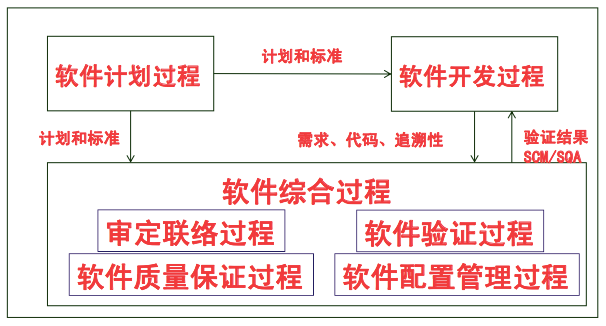


图 DO-178B/C软件生命周期方阵图

在整个软件生命周期过程中，产生了大量的软件生命周期数据，这些软件生命周期数据供审定局方来验证软件的研制开发已经满足了标准中的相应目标。DO-178B/C中例句了20种软件生命周期数据，包括：软件审定计划（PSAC）、软件开发计划（SDP）、软件验证计划（SVP）、软件构型管理数据（SCMP）、软件质量保证计划（SQAP）、软件需求标准、软件设计标准、软件编码标准、软件需求数据、设计描述、源代码、可执行目标代码、软件验证用例和规程、软件验证结果、软件生命周期环境构型索引（SCEI）、软件构型索引（SCI）、问题报告、软件构型管理记录、软件质量保证记录和软件完结总数（SAS）。

DO-178B/C根据软件失效条件将软件分为不同的等级，不同等级的软件需要满足不同数量的目标。标准对于每个过程都制定了相应的需要完成的目标，目标是作为审定方审定软件是否满足适航要求的最终依据。

DO-178B/C三个基本要素为上述介绍的过程、软件生命周期数据和目标，三者的关系如图所示。



图 DO-178B/C中基本要素的相互关系

### DO-178B/C标准中规定的可追溯性目标

DO-178B/C标准是面向目标的，标准中提及的目标是适航审定当局对于机载软件是否满足适航性要求的判断指标。针对软件开发过程中的软件需求过程、软件设计过程、软件编码过程和软件集成过程，DO-178B/C要求至少完成以下三个可追溯性目标：

1. 系统需求和高级需求之间的可追溯性

系统需求（SR）指的是机载系统对于某个软件功能的描述，是机载系统（非机载软件）在开发的过程中对软件提出的要求；高级需求（HLR）是对软件的某个功能的描述，是若干个需求的集合，因此在软件开发过程中软件的高级需求的完整性和正确性对软件的研制起着重要的作用。

系统需求与高级需求之间的追溯性关系连接了系统开发过程和软件开发过程，二者的双向可追溯性保证了系统开发层面要求的性能要求、系统功能和安全性方面的要求已经转换为了软件的高级需求。

（2）高级需求和低级需求之间的可追溯性

分配给软件的高级需求只是在一个抽象层面上表示了软件实现的功能，还需要进一步进行细化为可以进行编码的需求。低级需求（LLR）即为高级需求具象化之后的需求，对每一个小功能描述的更详细、对编码应该采用的语言和框架等描述的更清晰。但是在高级需求进行细化时，可能会引出衍生的高级需求（HLRN）[1]，即在系统需求可能只提供了一个大致的功能要求，但是部分高级需求并不能从系统需求追溯过来，因此需要提供一些衍生的高级需求对更具体的功能做支持；同理，有些高层需求描述的功能，能够与其追溯的低级需求并不能实现全部的功能描述，因此也需要提供衍生的低级需求（LLRN）。

高级需求和低级需求之间的追溯性关系是连接了抽象功能和底层实际编码需求之间的纽带，同时通过可追溯性保证所有的高级需求都必须被细化为了低级需求。

（3）低级需求和源代码之间的可追溯性

按照低级需求文档已经可以进行软件的编码工作等具体的软件开发，源代码（SC）则是低级需求的呈现结果。低级需求和源代码之间的追溯性关系是为了保证所有的软件低级需求都已经通过代码来实现。

图展示了以上的可追溯性关系。

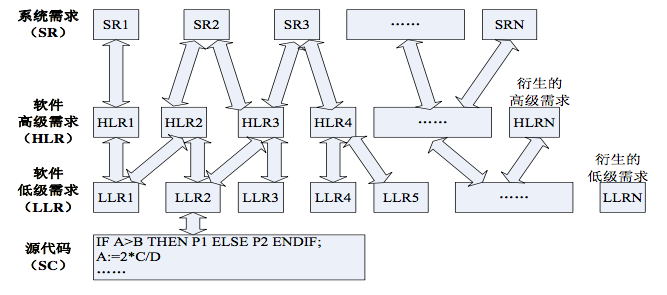


图 DO-178B/C中要求的追溯性关系

当然，除了以上在DO-178B/C的目标中严格规定的可追溯性之外，在软件的研发过程中还应该满足一些其他软件生命周期数据之间的可追溯性，比如软件需求与测试用例、测试程序、测试结果之间的追溯性等。

## 信息检索技术

信息检索技术是一个比较老的概念，指的是根据给定的查询项在一个数据集合上查找出相似的、相关的内容[Information Retrieval:A Survey]，并且结果按照相关度的大小排列。对于信息检索，大部分的工作是基于文本检索，即查询项和待查询集合的内容都是以文本形式存储，主要方法可以分为两类：基于统计检索和基于语义检索。

### 基于统计检索

基于统计的方法是根据查询语句和待查询集合的内容的统计度量来确定查询结果。将查询和待查询集合放在同一个统计空间中，对文本进行分词，将所有的内容基于统计信息进行表示，然后计算得到与查询项相关度内容。当前常用的基于统计检索的方法包括向量空间模型（Vector Space Model，VSM）和概率模型（Probabilistic Model）。

向量空间模型把文本信息用向量空间中的点表示，然后计算向量之间的距离来量化两个文本之间的相似度，最后通过对计算结果排序得到查询结果。文档向量构造最简单的方法是使用词袋模型（Bag of Words，BOW），即文档向量的长度是数据集中所有的词的数量，文档向量每一维的值为当前词在该文档中的出现频率或其他的一些特征，从而一个文档表示为一个向量。在计算文本间距离时，可以使用余弦公式等方法，得到两个文本之间的相似度值。还有一些方法通过对关键词加权重的方式表示文本向量[基于向量空间模型的短文本密文检索方法]。

潜在语义索引（LSI）也是基于向量空间的检索方法。LSI将单词和文档便是在同一个矩阵中，通过奇异值分解（SVD）的方法得到文本的主题。表示之后矩阵规模巨大、同时矩阵会过于稀疏，因此需要进行降维操作。虽然表示和计算过程都比较简单，但是也有如下几个缺点：（1）通过SVD生成的新矩阵解释性较差；（2）无法解决一词多义的问题；（3）和词袋模型一样，忽略了文章中单词的先后顺序。

概率模型是用多个随机变量之间非确定性的概率关系对解决信息检索问题，即将查询与文档集合中的文档同时出现的概率定义为二者的相似度，形式化表示如公式所示。同样在计算完成后，对相似度进行排序，然后设置阈值过滤得到某个查询对应的相关内容。

Sim（q, Di）=Pr(Di|q)=Pr(q|Di)Pr(Di)/Pr(q)

其中Di表示文档集合中第i个文档，q表示查询语句，公式使用贝叶斯公式对二者同时出现的概率进行求解。

### 基于语义检索

基于语义的方法是在一定程度上对查询句子进行句法和语义的分析，即通过其他的表现形式对文本的语义进行表示。由于当前大部分信息检索都是基于文本检索，因此基于语义的信息检索也和自然语言处理（Natrual Language Process，NLP）技术相关。使用句法分析的方法是借助文本的句式和一些词性信息对文本语义进行强化处理，如[基于子树匹配的文本相似度算法]中提到的使用子树加速对文本相似度的计算，并将子树与文本特征向量进行比较匹配，从而得到更准确的文本相似度；使用语义分析的方法有基于本体[\*\*]和基于外部语义词典（Hownet、Wordnet）[\*\*]等。

## word embedding介绍

Word2vec是Google的Tomas Mikolov[\*\*]等人提出的文本表示方法，将单词表示成word embedding的形式，是一种分布式表示（Distributed Representation）模型，反应了词的共现关系。在给定的语料库下，通过算法模型将词表示向量的形式，向量的维数可以由用户指定，通常为几十到几千不等，因此可以远远小于词典的大小，这样就避免了向量空间模型中的矩阵稀疏问题。同时，Mikolov等人还开发出了训练词向量的工具，训练效率非常高，非常适合在现在大数据集的应用中。

Word embedding基于分布式假说提出，即出现在相同上下文中的词的含义相近[\*\*]。训练word embedding的流程类似前馈神经网络，但是仅包含输入层、投影层和输出层三层结构，并且由于Mikolov等人对投影层和输出层做了优化，使得模型的训练效率更高。训练模型主要包括CBOW和skip-gram两种，CBOW模型的输入是特定词上下文相关词的词向量，输出是该特定词的词向量；skip-gram模型与CBOW相反，输入是特定词的词向量，输出是特定词对应的上下文词的词向量。CBOW和skip-gram模型分别如图和图所示。

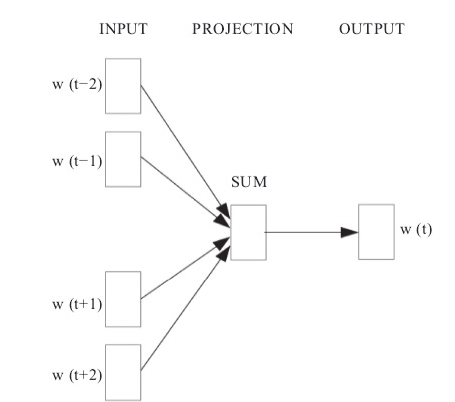


图 CBOW模型示意图

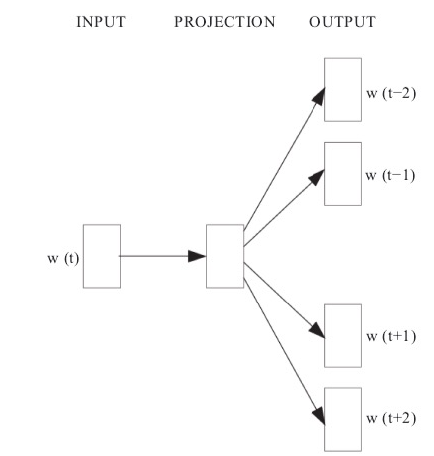


图 skip-gram模型示意图

## 软件工程领域学习排序算法

学习排序（Learning to Rank， LtR）算法作为一种监督或半监督的机器学习算法[16]，在信息检索、数据挖掘和自然语言处理等领域起着重要的作用。在软件工程领域，学习排序算法经常用在故障定位（Bug Localization/Fault Location）和数据重复检测（Duplication Detection）等方面。对于每个查询与候选文档的组合，我们能够抽取出若干特征，比如词汇相似度、语义相似度等，作为机器学习模型的输入。学习排序的流程如图所示，更加形式化的表示如公式所示。







其中表示查询语句，表示相应的候选文档集合，函数将查询-文档对映射为特征向量，是表示每个特征向量的权重矩阵，是排序方法。排序方法有三种类型：单文档方法（Pointwise）、文档对方法（Pairwise）和文档列表方法（Listwise）。

单文档方法的处理对象是单独的一篇文档，将其转化为特征向量后，该问题转化为回归问题，可以使用等级回归或分类算法来解决问题。该方法只考虑单个文档的绝对相关度，忽略了文档间顺序关系。文档对方法则考虑了文档间的关系，把任意两个文档组成的文档对（文档A，文档B）作为机器学习的输入，最后得到文档A是否应该排在文档B的前边，此时学习排序算法可以转化为二分类问题。该方法通过考虑文档两两之间的关系进行排序，因此比单文档方法的效果有所提升。文档列表方法的处理对象是每个查询对应的所有搜索结果列表整体作为一个训练实例，该方法直接优化算法模型输出的整体序列，因此其结果更能接近真实的文档序列。

除了排序算法，选择合适的特征作为模型的输入也对排序学习算法的效果有很大的影响。在软件工程领域，大多数使用学习排序算法的应用都选择词或者文本的相似度作为其中一个特征，比如[20]中作者使用了词表面信息相似度、通过API增强的词相似度和类名的相似度，[21]中作者选择文本相似度和上下文相似度作为学习排序算法的特征。值得一提的是，在计算以上相似度时，作者们都是用了余弦相似度。除此之外，特征可以分为两类：依赖查询的特征和不依赖查询的特征[21]。依赖查询的特征即与查询文本相关的特征，如相似度；不依赖查询的特征则与查询文本无关，仅仅反应候选结果的特点，如候选结果的词频[21]、故障修复频率[20]等。同时，在选择特征时，特征数量不应过大或过小，过大会导致过拟合问题，过小则会降低结果精确度。

表 排序方法的常用算法

|  |  |
| --- | --- |
| **排序方法分类** | **常用算法** |
| 单文档方法 | 回归：Subset Ranking 分类：SVM、McRank等级回归：Prank、Large Margin |
| 文档对方法 | IR SVM、Ranking SVM、RankBoost |
| 文档列表方法 | ListNet、AdaRank、SVM Map |

## 本章小结

本章首先介绍了适航领域中的DO-178B/C标准中对于软件生命周期的定义、标准中各个基本要素的关系以及标准中对于软件需求跟踪的一些严格的约束；然后介绍了信息检索技术中常用的技术，包括基于统计检索方法和基于语义检索方法；接下来介绍了word embedding的概念和训练方法；最后介绍了软件工程领域的学习排序算法的应用和常用排序算法。本章通过对适航领域软件特点的分析和需求跟踪技术研究中相关的技术的介绍，为下一章适航领域软件需求跟踪算法模型的构建提供了理论基础和支撑。

# 适航领域软件需求跟踪算法模型构建

本章详细的介绍了软件需求跟踪模型的构建。首先介绍了需求跟踪模型的整体框架，并描述了构建模型时应考虑的问题。接下来从模型包含的几个部分出发，详细介绍了数据预处理阶段、计算文本相似度阶段和学习排序阶段三个阶段的构建过程。最后对模型构建过程进行总结。

## Tr-WELR模型框架

根据需求跟踪技术流程，文本相似度算法的选用是需求跟踪模型性能的关键。模型的整体流程如图所示，包括数据预处理阶段、计算文本相似度阶段和学习排序阶段。

（1）数据预处理阶段包括适航软件文档的预处理和词向量的训练，其中适航软件文档的预处理包括文档的脱密处理和文本预处理。基于适航文档的重要性和机密性，在处理适航软件文档之前应该先进行脱密处理，一般使用的脱密方法为将重要的中文名词使用相应汉字拼音首字母表示，如保密数据“北航纠错系统”可以写作“BHJCXT”。对具体的软件文本预处理部分在实验部分详细描述。由于维基百科数据的数据量庞大，因此为了保证训练的词向量的专业性，选择了仅使用抽取软件相关文章进行训练，具体操作过程在第四章实验部分详细描述。

（2）预处理之后，将软件文档映射为词向量，然后输入到计算文本相似度算法中，该算法命名为WQI。在算法中，我们将信息检索中查询的概念与待建立跟踪关系的需求文本等价，即每个需要建立跟踪链接的软件需求文本是信息检索中的一个查询语句。算法对每一个查询语句进行文本相似度计算，最终每个查询都得到一个候选的排序列表，排在最前边的为与相应查询相关度最高的软件制品。

（3）最后，通过将排序列表引入学习排序算法模型中，利用机器学习分类算法将排序后的列表进行二次处理，最终得到跟踪链接的最终列表和针对每个数据集的一个跟踪链接预测模型。

接下来将详细介绍Tr-WELR模型的三个部分。

图 Tr-WELR模型流程图

## 加权策略和查询扩展

### 加权策略选择

处理之后的文本数量仍然规模较大，同时为了区分句子中的单词的重要程度，并且简化之后的计算，我们选择对句子中的单词进行加权处理。在这里我们使用TF-IDF的策略来计算，如公式所示。当一个词或短语在本文档中出现的频率较高，而在其他文档中出现频率较小时，则可以认为这个词或短语具有较强的类别区分能力，即能够比其他词更有代表当前文档的能力。TF（词频，Term Frequency）表示某个词在当前文档中的出现频率，IDF（逆文档频率，Inverse Document Frequency）的思想是：如果包含某个单词或短语的文档数量越少，则该值越大，即表示当包含某个词或短语的文档数越少，该词在当前文档中所占的权重越高。







其中指单词或短语在文档中的词频，分子是词在文档中出现的次数，分母是文档中单词的总数。指单词或短语在文档集合中的逆文档频率，分子表示文档总数，分母表示出现过此单词或短语的文档数，一般为了防止分母为0，会给该文档数加一，最后对比值取对数。

### 查询扩展算法

在论文中，计算两个词之间的语义相似度我们首先将词表示为词向量的形式，然后使用cosine相似度计算，如公式所示。



其中和分别表示词和的词向量，和分别表示两个词的词向量的长度。

合适的查询扩展方法和加权策略能够提升信息检索任务的性能[22]。查询扩展的流程如图所示。对某个文档进行查询扩展，是对文档中的各个词或词组进行扩展，步骤如下：

（1）首先计算TFIDF权重，然后根据TFIDF进行排序，选择前topn%的单词或词组进行扩展，组成集合，其中topn%参数设置为经验值0.3。

（2）对以上选择出的前topn%的单词或词组进行扩展，使用单词语义相似度计算公式，选择出相似度大于阈值的单词或词组，每个被扩展的词组成集合，形式化的表示如公式所示。其中参数同样设置为经验值0.7。





图 查询扩展流程图

## 改进的文本语义相似度算法

在论文[11]中，文本语义相似度的计算有三个步骤：1）计算词之间的语义相似度；2）计算词与文本之间的语义相似度；3）计算文本之间的语义相似度。词与词的相似度计算如公式所示，词与文本的相似度计算即为词与文本中所有词相似度的最大值，如公式所示。文本与的相似度为文本中的每个词与文本的相似度的和，然后做归一化处理，如公式所示。







其中表示文本中与文本的语义相似度不为0的单词或词组集合。

以上即为论文[11]中提出的方法，我们称之为W2V方法。相对于该方法，我们做出如下改进：

（1）在计算词与文本的相似度时，添加上查询扩展，于是重写了公式x，如公式所示。添加上查询扩展后，同时还设置了参数来分配原词的相似度和扩展的词的相似度之间的权重，该值的取值范围为，在实际的计算过程中，该值在0.5到0.9之间，以0.01的步长增长，最后取能够得到最好效果的值，因此该参数值根据不同数据集的不同而不同。如果当前单词没有扩展词，则使用与论文[11]中相同的方法，即使用公式x计算。



其中表示查询扩展项，即当计算词与词之间的相似度时，如果该词具有扩展词，则扩展项使用词的扩展词集合中的词与词之间的相似度之和，然后做归一化处理，保证的值的范围在0到1之间。举个例子，单词“technique”的扩展词有“technology”、“method”和“approach”等，当计算单词“technique”与句子“The basic requirement of planes is safety.”的相似度时，不仅仅计算“technique”与句子的相似度，同样会计算其扩展词“technology”、“method”和“approach”与该句子的相似度，最后使用比例参数将两者结合起来。

（2）在计算文本之间的相似度时，原方法的公式是不对称的，即对于同样的一对文本和文本，不同的输入顺序会产生不同的结果。为了保证一致性，以及结果与语句长短的无关性，本方法增加了一个限制，即在输入时使文本的长度长于文本。增加该限制主要有两个原因：第一是使得该公式是对称的，当计算任意一对文本的相似度时，结果与二者的前后顺序无关；第二是当较短文本是较长文本的子集时，避免较长文本中的其他词汇被忽视。第二个原因是一种特殊情况，举个例子：文本和文本分别是“Airworthiness is necessary.”、“This document emphasizes that airworthiness is necessary.”，在使用公式x计算时，文本中的每个词在文本中都出现了，因此在使用公式y计算词与文本的相似度时，文本中的每个词与文本中的词的相似度的最大值均为1，此时文本中的其他词被忽略掉了。为了避免这种情况，限制两个文本的前后顺序是重要的。

## 学习排序算法

通过上述的文本语义相似度计算之后，对于每一个查询都可以排序生成一个排序列表，即候选列表，设置相应的阈值，即可产生相应的需求跟踪关系。但是，此时仅仅使用了文本语义相似度这一个特征，还有一些没有使用的特征。综合使用其他的特征可以提升结果的准确率，因此本方法使用学习排序算法来进一步对已有的排序列表进行处理。以下将从两个方面描述学习排序算法：特征选择和排序算法。

### 特征选择

本文选用了五个特征作为学习排序模型的输入，五个特征可以被分为两组：依赖查询的特征和不依赖查询的特征，如表所示。前三个特征为依赖查询的特征，这几个特征强调查询与候选结果之间的关联关系；后两个为不依赖查询的特征，更强调候选结果文本自身的特点。

表 学习排序算法选择的特征

|  |  |  |
| --- | --- | --- |
| **特征** | **描述** | **是否依赖查询** |
| 语义相似度 | 使用WQI方法计算的文本语义相似度 | 是 |
| 广义Jaccard系数 | 文本的上下文相似度，文本向量由词向量取平均值表示 | 是 |
| IDF之和 | 候选结果文本上的当前查询中所包含的词或短语的IDF值之和 | 是 |
| 关键字数量 | 候选结果中关键字数量，通过计算TFIDF然后排序和取阈值得到 | 否 |
| 文本长度 | 候选结果文本的长度 | 否 |

（1）语义相似度，通过提出的语义相似度算法WQI计算；

（2）广义Jaccard系数，通过计算文档向量的关系表示两个文档上下文相似度，在本文应用中即为查询语句和一个候选文档之间的上下文相似度，如公式所示。



其中，向量和向量均为文档向量，此处文档向量使用词向量的平均值表示。和分别表示向量和向量的数值大小。

（3）IDF之和，在候选结果文本上中计算查询中的词或短语的IDF值，并求和，用来表示当前查询在某一候选结果上的重要程度，如公式所示。



其中为查询中的单词或短语，表示在查询结果中的idf值。

（4）关键词数量，表示查询候选结果中关键词的数量，关键词的界定使用TFIDF的值，计算过程与加权策略中介绍的过程相同。

（5）文本长度，表示查询候选结果的长度，在一定程度上体现该结果的有效程度。

### 排序算法

正如第二章中所提到了，单文档方法（Pointwise）仅考虑了单个文档与查询的绝对相关度，忽略了文档间的顺序关系；文档对方法（Pairwise）考虑了任意两个文档之间的相对顺序关系，相比单文档方法的效果更好；文档列表方法（Listwise）需要考虑每次对查询候选结果列表，当文档数量较大时，需要考虑的数量较大，相对而言没有文档对方法的效率高。综合以上原因，Tr-WELR方法模型中使用了文档对方法，并采用成熟的IR SVM方法。

IR SVM的基本思想是将排序问题转化为SVM的分类问题，将两个样本和表示成一个训练样本，然后使用SVM模型进行学习和求解。由于IR SVM是对Ranking SVM的改进，下面首先对Ranking SVM进行介绍，然后再详述IR SVM对Ranking SVM的改进。

#### Ranking SVM

Ranking SVM算法的思想是首先对查询和文档对进行分类，然后将训练得到的分类器用在排序任务中[7 Large Margin rank boundaries for ordinal regression]。在排序应用中个，假定有两组查询对应的文档集合，每组查询有三个等级，分别是等级1、等级2和等级3。举例说明，在第一组查询结果中有三个对象、和，分别属于等级1、等级2和等级3，如图所示。图中的向量即为，之前公式（LtR公式）中的权重向量。但是，此时对于每一个查询得到的候选文档集合，都有一个各自的排序向量，在应用中非常麻烦。因此，对同一组查询结果集合中的不同等级的对象的特征向量进行组合，形成新的特征向量，即将对象映射到另一个向量空间，然后根据另一个向量空间的向量进行排序。如果排序算法比较好，那么在同一组查询中的候选结果区分度也会比较好。比如将上述的三个对象重新组合为：、和，并且给这些对象重新打标签，如将、和标记为负相关，相应的、和标记为正相关，如图所示，即可将以上的排序问题转化为二值分类问题，可以通过训练线性SVM分类器对上述新的向量空间的向量进行分类，进而可以计算得到同一组查询结果集中的不同等级的向量的前后词序。

训练数据表示为，其中为，每一个训练实例由两个特征向量表示，标签由表示，且的取值为+1和-1，分别表示正相关和负相关。Ranking SVM的最优化问题可以形式化的表示为公式。





其中为松弛变量，是训练实例的数量，是第二范式，把松弛变量代入公式中，可以得到：



其中，加和的第一项表示hinge loss, 第二项为正则项，防止过拟合。



图 排序问题示意图



图 将排序问题转化为分类问题示意图

#### IR SVM

IR SVM是Ranking SVM在信息检索领域的一个改进[15 Adapting ranking SVM to document retrieval]。Ranking SVM在信息检索领域的不足之处包括两个方面：

（1）Ranking SVM是将排序问题转化为分类问题，在学习过程中使用了0-1分类损失函数。在信息检索任务中，最终结果列表中排在前面的对检索效果的影响更大，而Ranking SVM对待每个不正确的相对顺序都一视同仁。举例说明，有三个等级，其正确的排序顺序应该为等级1、等级2、等级3，则等级3>等级2和等级3>等级1都是错误的相对顺序，但是二者对Ranking SVM的训练过程造成的影响是相同的，显然，这与实际的排序过程有一定的误差。

（2）另一方面，Ranking SVM同等对待不同查询下的结果对。举例说明：每个文档使用等级来表示，两个查询的结果如表所示。对于查询一，在转换向量空间时可以有2个等级3-等级2的文档对，4个等级2-等级1的文档对，2个等级3-等级1的文档对，共8个文档对可供训练；对于查询二，则有两个等级3-等级2的文档对，八个等级2等级1的文档对，四个等级3-等级1的文档对，共有14个文档对可供训练。虽然两个查询得到的结果中等级结构相同，但是由于数量不同，查询二对Ranking SVM模型的影响会比查询一大，因此最后的结果会有偏差。同时，这也与信息检索任务中所要求的“每个查询的重要性等同”[\*\*]是不相符的。

表 排序列表举例

|  |  |
| --- | --- |
| **查询** | **结果** |
| 查询一 | 五个文档：等级3、等级2、等级2、等级1、等级1 |
| 查询二 | 七个文档：等级3、等级2、等级2、等级1、等级1、等级1、等级1 |

由于Ranking SVM以上两个方面的不足之处，IR SVM将二值分类问题改进为代价敏感的分类问题（Cost Sensitive Classification），即对来自不同查询的文档对，或者不同等级的文档对设置不同的损失权重。由于在排序结果列表前面的结果比后面的结果的重要程度高，在计算时会对在前边出错的文档对加大损失函数的权重，即越在文档列表前边出错，所需要付出的代价越高；出于对上述Ranking SVM第二点的不足的改进，在计算时排序结果中文档集较少的情况下，如果有顺序出错的文档，也会加大损失函数的权重，减少查询本身对查询效果的影响。

讲IR SVM算法。。。。。。

## 本章小结

本章首先给出了基于word embedding和学习排序算法的适航领域需求跟踪算法模型框架，并对框架中包含的几个部分进行了简单介绍。接下来详细介绍了算法模型中改进的部分，其中包括：在计算文本语义相似度时提出的在word embedding的基础上添加加权策略和查询扩展，并一次为基础改进了文本语义相似度的计算算法。鉴于信息检索技术对软件需求跟踪关系的恢复结果对数据的特征使用较少，提出了使用学习排序算法对结果的精度做进一步的提升。本章对以上算法和实际使用中使用的算法做了详细的介绍，对模型的构建过程做了详细的描述。

、

# 适航领域软件需求跟踪算法模型验证

为了验证和评估第三章所提出的模型Tr-WELR的有效性和性能，本章将通过实验对其验证。首先对验证模型所需数据集的来源和相关特征进行详细描述，然后介绍实验设置和实验具体内容，接下来对文档预处理进行详细描述，然后介绍需求跟踪算法模型的评估指标，最后给出了详细的实验的结果和分析。

## 数据准备

数据获取需要的步骤如图所示，包括（1）相关人员对需求文档、设计文档、代码文件、测试文档、需求变更文档、审定文档和可能有的需求跟踪矩阵文档等项目相关文档的收集，对不存在需求跟踪矩阵或其他形式的需求跟踪文档的情况，还需要相关专家对部分需求的跟踪关系进行标注；（2）对数据进行脱密处理，常用的脱密方法如表所示[军校学报稿件的脱密加工方法与技巧]；（3）对文档进行自动解析，根据文档的结构和标题，将不同生命周期的数据抽取出来，将半结构化数据转为结构化数据，同时仍然需要录入人员对数据的分类情况进行确认；（4）确认完成，将数据存入数据库中。

图 数据获取过程示意图

表 常用脱密方法

|  |  |
| --- | --- |
| **脱密方法** | **详情** |
| 彻底删除法 | 对工艺技术和关键数据，彻底删除 |
| 模糊处理法 | 对特定的装备，可以省去装备具体名称，使用“某型”等或其他模糊名称；对某个具体数字，可以使用虚数表示 |
| 替换取代法 | 将保密的某个装备表示为“典型装备”或其他特定名字 |
| 数据示意法 | 对于具体某个装备性能的分数或参数，使用合理取值范围内其他数字表示 |

本文选择了在跟踪恢复领域经常使用的五个公开数据集来验证提出的算法模型的有效性和性能。数据集可以在CoEST网站上获取到，包括：CM1-NASA、GANTT、eTOUR、iTrust和EasyClinic，其中CM1-NASA是美国国家航空航天局的基础项目CM1的一个子集；GANTT是一个使用甘特图来管理项目流程的软件；eTOUR是由Salerno大学使用java语言开发的一个电子旅游导航软件；iTrust是一个用于记录医药信息的java web系统；EasyClinic是用在医疗管理的java软件项目，同样由Salerno大学开发。每个数据集的具体情况如表所示。CoEST网站不仅提供了测试数据，而且对每一组测试数据都给出了专业的跟踪链。使用CoEST数据进行实验的优势包括：1）给验证模型的工作带来了极大的便利；2）保证了结果的准确性；3）减少了在学习排序阶段对数据的标注过程。

表 实验数据集

|  |  |  |
| --- | --- | --- |
| **数据集名称** | **包含的内容** | **词条（Token）数量** |
| CM1-NASA | 22个高层需求、53个底层需求、45个跟踪链接 | 5767 |
| GANTT | 17个高层需求、69个底层需求、68个跟踪链接 | 2080 |
| eTOUR | 58个用例、116个代码类、308个跟踪链接 | 97452 |
| iTrust | 131个用例、367个代码类、534个跟踪链接 | 123501 |
| EasyClinic | 30个用例、20个UML交互图、63个测试用例、47个类描述信息、1257个跟踪链接 | 21882 |

在确定软件数据集之后，还需要生成训练词向量的语料库。Siwei L.等人[24 How to Generate a Good Word Embedding?]指出在训练词向量时，语料的领域相关性比语料库的大小更重要，并且语料的领域性越强，词向量的表示效果越好。当然，在相同的领域，语料库越大效果越好。为了保证领域相关性，我们首先下载了一份软件术语表，然后去掉软件术语表中单个词的短语，接下来对每个软件术语添加上“software”，最后使用处理过的软件术语表去过滤维基百科数据集（wiki dumps），最终得到一个包含有43，443，648个单词的264M的软件相关语料库。具体步骤如图所示。



图 获取训练词向量语料库流程

## 文档预处理

实验中文档根据类型可以分为两类：（1）文本文件，包括高级需求文档、底级需求文档、用例文档、UML交互图文档、测试用例文档、类描述文档等；（2）源代码文件。具体的文档预处理流程如图所示。

第一步，设置领域词典。使用现有的中文分词工具对中文软件文档进行分词时，并不能对很多领域专有名词进行区分，我们首先通过OCR（Optical Character Recognition，光学字符识别）将pdf版本的《XXXX词典》中的汉语和对应的英文解释提取出来，然后汉语部分作为领域专有名词，并将汉语和英文部分存入数据库中。

第二步，对中文软件文档进行分词和翻译。步骤如下：（1）使用结巴分词工具，并在分词时加入自定义词典，即第一步中的领域词典对文本进行分词；（2）去掉中文分词结果中的中文停用词；（3）分词得到的结果将优先使用领域词典对单词进行翻译，当领域词典中该词汇不存在时，使用有道云翻译接口进行翻译；（4）最终于的分词和翻译结果存入数据库中。

第三步，对分词并翻译之后的文本和代码文件进行以下处理：（1）以空格为分隔符对文本和代码内容进行分割；（2）去掉标点符号、数字和英文停用词。另外，在处理翻译后的文本时，使用的是nltk[34 Natural Language Processing with Python]，并给英文停用词列表中加入了一些数据集中无用的词；在处理代码文件时，我们将java和C语言中的关键字加入了停用词列表；

第四步，根据驼峰命名法的规则对复合名词进行分解。需要注意的是，这里只对不在软件术语表中出现的复合名词进行分解，即只对由开发者命名的复合名词进行分解。举例说明，单词“DefaultMutableTreeNode”将被分解成四部分：“Default”、“Mutable”、“Tree”和“Node”。

第五步，将所有的单词都转为小写形式。需要注意的是，在本文提出的方法中并不需要对单词进行词干提取或词形还原。

最后，将以上的结果存入数据库中。



图 文档预处理流程

## 实验设置

正如第三章提到的，Tr-WELR模型中计算部分包括使用WQI方法计算文本语义相似度阶段和学习排序阶段。本文设置了三组实验，第一组实验是使用LSI[30 Can LSI help reconstructing requirements traceability in design and test?]方法和公式x所示的中计算文本相似度的方法，分别与WQI方法做对比，用来验证提出的改进的文本语义相似度算法对结果的提升效果；第二部分实验是对使用学习排序对列表进行处理的前后进行对比，用来验证学习排序算法对领域软件需求跟踪算法性能的提升效果；第三部分实验时对比本文提出的Tr-WELR模型和当前国际领先水平的方法AML[35 Information retrieval and spectrum based bug localization: better together]，用来验证本文提出的Tr-WELR模型的有效性和效果。

在本文中，LSI方法指的是使用潜在语义索引方法恢复软件跟踪链接的方法，该方法已经被证明是一个在跟踪恢复领域成熟的方法[7 Software traceability: trends and future directions.]。该LSI方法步骤分为六步[30 同上]，分别是：（I）定义一个潜在的跟踪模型；（II）在数据集上使用跟踪模型自动识别出文档所属生命周期类型；（III）预处理文档；（IV）重建跟踪链接；（V）选择相关的跟踪链接；（VI）可视化跟踪链接。

表 对比方法

|  |  |
| --- | --- |
| **实验** | **对比方法** |
| 实验一 | LSI、W2V |
| 实验二 | 学习排序前后对比 |
| 实验三 | AML |

本文使用主题模型工具gensim[33Software framework for topic modelling with large corpora. ]在生成的语料库上训练词向量，同时选择word2vec训练模型中的CBOW模型，设置窗口大小为5，词向量的维度为200。

本论文中所有的实验都在相同的实验环境下进行，实验所用的机器为八核英特尔i7处理器，运行内存为8G。

## 模型评估指标

为了评估模型的性能，本文使用精确率、召回率、F测度、MAP（Mean Average Precision）和MRR（Mean Reciprocal Rank）为评估指标。在描述评估指标之前，记：

·Ntp为检索列表中与查询项有关联的文档数，即可以与查询的需求建立跟踪关系的文档的数量。

·Nfp为检索列表中与查询项无关联的文档数，即与查询的需求无法建立跟踪关系的却与之建立跟踪关系的文档数。

·Ntn为数据集中与查询项没有关联的文档，同时也没有在检索列表中出现的文档数。

·Nfn为数据集中与查询项有关联的文档，但是不在检索列表中的文档数。

下面给出在软件需求跟踪领域中，常用的模型性能评估指标定义。

精确率（Precision）检索列表中出现的文档中与查询项有关联的文档占检索列表文档总数的比。

PRE = Ntp / (Ntp + Nfp)

召回率（Recall）检索列表中与查询项有关联的文档数占数据集中与查询项有关联的文档数的比。

REC = Ntp / (Ntp + Nfn)

F测度（F-measure）好的模型应该同时有好的精确率和召回率，但是精确率和召回率有时候是矛盾的，即当精确率较高时，召回率低；当召回率较高时，精确率低。F测度为精确率和召回率的调和平均值，当F测度值较高时精确率和召回率的结果都较高。

F=2\*PRE\*REC/(PRE+REC)

平均准确率（Average Precision）平均准确率是对单个查询而言，即对单个查询的结果集的准确率的平均值。

AP=1/m \* 求和i->m i/ranki

MAP（Mean Average Precision）该值为数据集合的平均准确率，所有单个查询的平均准确率的平均值。举例说明：软件数据集中共有两个需求，req1和req2，两个需求对应的可以建立跟踪关系的文档分别有4个和5个。需求req1对应的文档的排名为1、2、4、7，需求req2对应的文档的排名为1、3、5，则对于需求req1的平均准确率为（1/1 + 2/2 + 3/4 + 4/7）/ 4 = 0.83，对于需求req2的平均准确率为（1/1 + 2/3 + 3/5 + 0 + 0）/ 5 = 0.45，那么该软件数据集的MAP = （0.83 + 0.45）/ 2 = 0.64。

MAP=1/n \* 求和i->n APn

MRR（Mean Average Precision）与MAP所关心所有正确的文档位置不同，MRR仅关心第一个正确的文档的位置。对每个查询对应的候选文档列表，找到每一个集合中第一个正确的文档的位置，然后取倒数，再对所有集合中的倒数取平均值，如公式所示。

MRR=1/n \* 求和i->n 1/ranki

其中ranki为集合中第一个正确的文档的位置。举例说明：对于软件数据集共有两个需求，分别是需求req1和需求req2，需求req1建立跟踪关系的候选文档集合列表为doc1、doc2、doc3，其中doc2是与需求req1有跟踪关系的文档，需求req2对应的候选文档集合列表为doc4、doc5、doc6、doc7，其中doc6是与需求req2有跟踪关系的文档，则MRR = （1/2 + 1/3）/ 2 = 0.53。

## 实验结果与分析

这一节将详细描述三组实验的实验结果，根据结果对提出的适航领域软件需求跟踪算法模型的有效性和性能进行分析，并对一些性能提升点和异常情况进行详细的解释说明。

### 第一组实验：使用WQI算法

本组实验分别使用LSI方法、W2V方法和WQI算法对七组数据进行跟踪链接的恢复，并使用精确率、召回率和F测度进行模型评估，实验结果如表所示。表中PRE和REC分别表示精确率和召回率。HL、LL、UC、CC、ID、TC分别表示高层需求文档、底层需求文档、用例文档、类文件、UML交互图文档和测试用例文档。

在结果中可以看到，后两种使用词向量的方法W2V和WQI平均来说比LSI的效果更好。其中，WQI算法和LSI方法相比，WQI算法比LSI在精确率上相对提升了33.3%，在召回率上相对提升了24.5%；WQI算法和W2V方法相比，WQI算法比W2V方法在精确率上相对提升了6.6%，在召回率上相对提升了17.2%。

表 三种方法的实验结果

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Datasets** | | **LSI** | | **W2V** | | **WQI** | |
| **PRE** | **REC** | **PRE** | **REC** | **PRE** | **REC** |
| CM1-  NASA | HL→LL | 0.127 | 0.41 | 0.262 | 0.217 | 0.371 | 0.329 |
| GANTT | HL→LL | 0.286 | 0.332 | 0.278 | 0.418 | 0.255 | 0.563 |
| eTOUR | UC→CC | 0.077 | 0.221 | 0.098 | 0.332 | 0.088 | 0.415 |
| iTrust | UC→CC | 0.009 | 0.45 | 0.192 | 0.363 | 0.198 | 0.322 |
| Easy  Clinic | UC->ID | 0.259 | 0.833 | 0.338 | 0.75 | 0.342 | 0.806 |
| UC->TC | 0.45 | 0.755 | 0.522 | 0.867 | 0.499 | 0.867 |
| UC->CC | 0.317 | 0.503 | 0.215 | 0.677 | 0.232 | 0.76 |

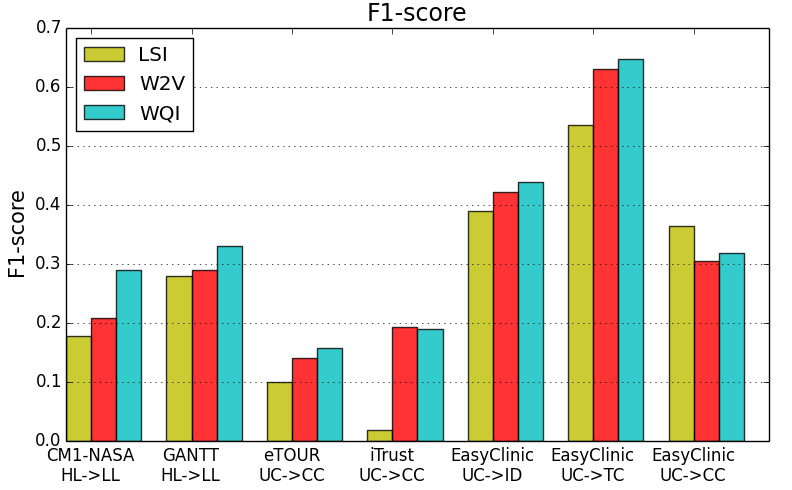


图 LSI、W2V和WQI三种方法在七组数据上的F测度

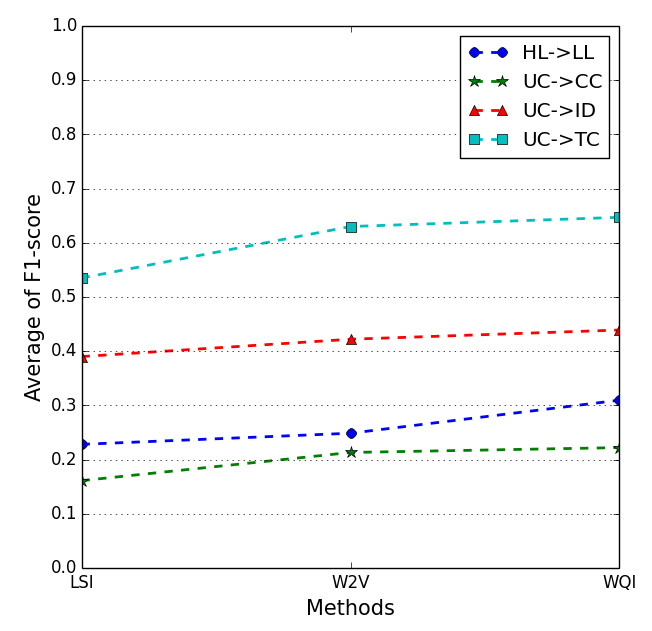


图 LSI、W2V和WQI三种方法在不同类型的跟踪链接上的平均F测度

图展示了三种方法在七组数据上F测度的平均值，显示了WQI算法比LSI方法和W2V方法在大多数数据集上效果都更好。

从表中可以看出，WQI算法在七组不同类型的跟踪链接上，相比其他方法都有普遍的性能提升。跟踪链接能够根据软件文档的类型分为两类：一类是文本对文本（t2t），另一类文本对源代码（t2c）。表中HL->LL、UC->ID和UC->TC都属于t2t，UC->CC属于t2c。同样，在图中也能看出，WQI算法对两类跟踪链接的F测度都有提升。对于以上性能的提升，可以做如下解释：对于t2t类型的跟踪链接的性能提升，表示本文提出的改进的文本语义相似度算法WQI在软件需求跟踪任务是可行的，且效果不错；对于t2c类型的跟踪链接的性能提升，表示在实验之前对代码文件的预处理操作是合理的，且通过词向量表示时能够相对准确的表示其在代码中的实际含义。

相比W2V方法，WQI算法引入了TFIDF加权策略和查询扩展技术，并且改进了计算的过程。从结果上来看，WQI算法所做的改进提升了结果的精确率和召回率，即提升了需求跟踪任务的性能。查询扩展技术使用和查询项相关或相似的单词或短语对查询项进行扩展，这也就在语义空间上丰富了查询项的含义[25 Using word embeddings for automatic query expansion.]。从另一个角度来说，因为每一个词向量包含词的频率特征和上下文的信息，因此一组相似的词必定比一个特定的词包含的信息多。另外，权重策略TFIDF不仅仅能够减少句子中的一些噪声信息，也通过抽取句子中的关键词减少了待扩展到词，即提升了精度又提高了算法执行效率。

综上所述，相比LSI方法和W2V方法，文本语义相似度计算算法WQI能够在软件需求跟踪任务中得到更好的效果，即生成的排序列表的质量更高。

### 第二组实验：使用学习排序算法

学习排序算法将利用查询和排序列表的一些特征去学习，预测新的需求跟踪链接。本文使用学习排序算法去提升需求跟踪任务的精确度。为了验证学习排序算法是否可以让软件需求跟踪任务取得更好的效果，本节做了实验去对比使用学习排序算法前后对结果的影响。本文使用了IR SVM作为学习排序的排序算法，同时使用MRR和MAP作为模型评估标准。学习排序算法使用之前介绍的五个特征去生成排序模型，同时做了十折的交叉验证，并且最终取这些验证的平均值作为最终结果。

图展示了执行学习排序算法前后的MRR和MAP指标的值。和执行学习排序算法之前的结果相比，执行算法之后MRR和MAP的平均值都有所提升，具体来说，MRR的平均值提升了3.5%，MAP的平均值提升了15.9%。MAP指标的显著提升，表明IR SVM算法能够提升候选结果列表的整体排序水平和整体结果的精度；MRR指标的提升，说明在候选结果列表中首次出现正确的文档的位置更靠前。从另一个角度来说，使用学习排序算法后，模型能够提供更准确的跟踪链接。

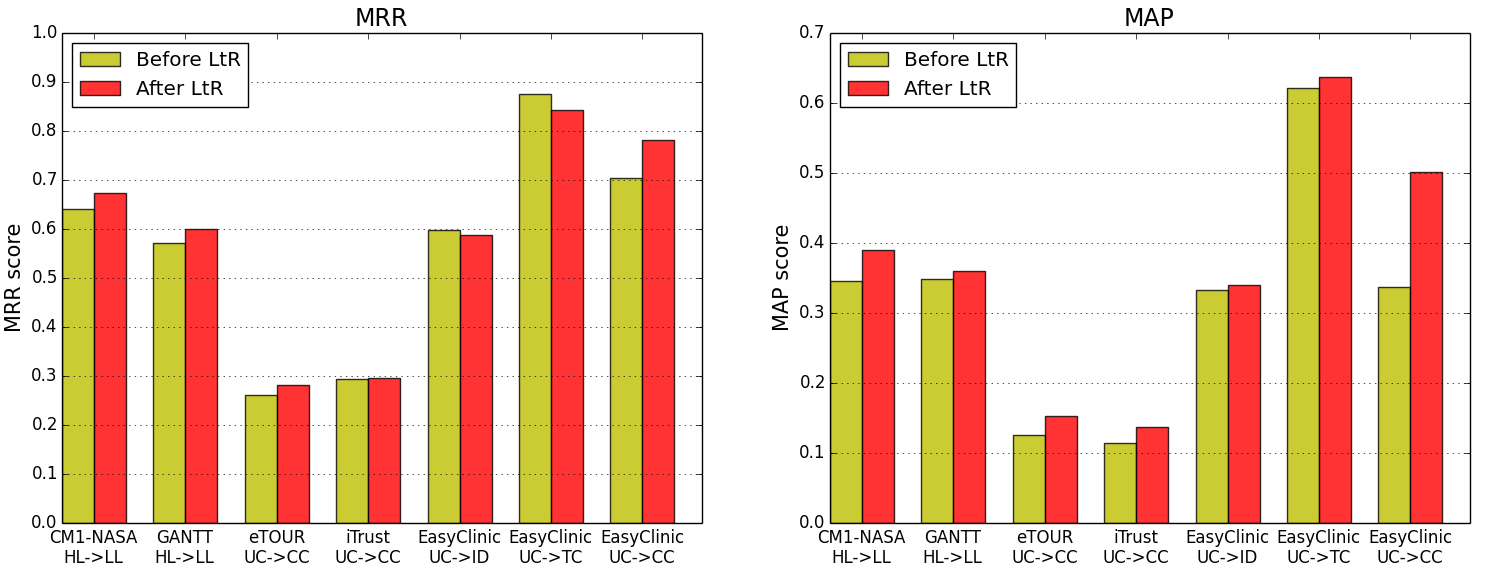


图 学习排序算法执行前后的MRR和MAP比较

### 第三组实验：Tr-WELR模型对比AML算法

AML[35]是一个新的多模型的故障定位技术，在多个软件项目上去的非常好的效果，包括AspectJ、Ant、Lucene和Rhino等。AML方法结合了信息检索技术和program spectrum并整合了三种不同类型的相似度去得到最终结果。同时，该方法在故障定位方面是通用的且有效的。本节将在eTOUR、iTrust和EasyClinic三个数据集上对比Tr-WELR模型和AML算法在需求跟踪任务上的性能和效果。选择这三个数据集是因为这三个数据集上都有文本数据和代码数据，能够更全面的显示算法的性能。在本组实验中，我们依然使用MRR和MAP指标来对模型进行评估。

像论文[35]中的提到的一样，本文将一次计算AMLtext、AMLspectra和AMLsuspword，最后使用公式将三者整合在一起。



其中xi表示一组特定用例，θ是参数向量[α, β,γ]，参数通过论文中提到的概率学习方法调节。

表和表分别列出了AML和Tr-WELR两个模型的MRR和MAP指标的值。从两个表中可以看出Tr-WELR模型比AML模型在MRR指标上相对提高了30.8%，在MAP指标上相对提升了11.8%。

表 AML和WELR的MRR指标比较

|  |  |  |
| --- | --- | --- |
| **Datasets** | **AML** | **WELR** |
| eTOUR | 0.196 | 0.281 |
| iTrust | 0.247 | 0.296 |
| EasyClinic | 0.605 | 0.781 |

表 AML和WELR的MAP指标比较

|  |  |  |
| --- | --- | --- |
| **Datasets** | **AML** | **WELR** |
| eTOUR | 0.134 | 0.153 |
| iTrust | 0.122 | 0.137 |
| EasyClinic | 0.460 | 0.501 |

## 运行时间效率

表显示了Tr-WELR模型中在不同数据集上的计算部分的运行时间，其中计算部分包括文本语义相似度计算阶段和执行学习排序算法阶段。表中Mean R.T.(sec)表示在特定数据集上恢复一组跟踪链接需要的平均运行时间，Retrieval Nums指的是在每个数据集上对任意一个查询项需要检索的文档数，Mean Documents’ Length指的是每个数据集的平均文档长度。根据结果我们可以推断出平均运行时间和检索数目和平均文档长度都是正相关的，即需要检索的文档数越多，平均文档长度越长，需要的计算时间越长。

表 Tr-WELR模型计算部分的运行时间和相关信息

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Datasets** | | **Mean R.T.(sec)** | **Retrieval Nums** | **Mean Documents’ Length** |
| CM1-  NASA | HL→LL | 2.44 | 53 | 339 |
| GANTT | HL→LL | 2.05 | 69 | 107 |
| eTOUR | UC→CC | 49.42 | 116 | 1766 |
| iTrust | UC→CC | 79.25 | 367 | 2867 |
| Easy  Clinic | UC->ID | 18.09 | 20 | 604 |
| UC->TC | 27.00 | 63 | 604 |
| UC->CC | 29.78 | 30 | 604 |

## 本章小结

本章是本文的实验部分，目的是对本文所提出的软件需求跟踪模型Tr-WELR进行验证。首先介绍了数据准备流程和实验所用到的数据集特征和来源，然后根据文本种类的不同分情况介绍了文档预处理过程，接下来对实验中的对比方法、训练词向量的语料库和实验环境做了介绍，并给出了模型的评估标准，最后对实验的结果和现象进行分析，给出结论。实验证明，在多组数据集和多个不同类型的跟踪链接上，提出的WELR模型能够较好的完成恢复跟踪链接的任务，主要原因有两方面：1）和提出的基于word embedding的文本语义相似度算法在软件文档上使用且准确率有所提高；2）使用学习排序算法使得跟踪结果的精度提高。

# 适航领域软件需求跟踪算法模型原型系统

## 系统需求分析

## 开发环境

## 系统总体设计

## 模块设计与实现

## 本章小结

# 总结与展望

本章介绍网络异常检测系统的实验工作，基于实际校园网流量验证该系统的异常检测功能，并对异常时段的流量进行分析。

## 总结

## 展望

# 参考文献



# 攻读硕士学位期间取得的学术成果

1. Qin Xi, Xu Tongge, Wang Chao. DDoS Attack Detection Using Entropy and Clustering Technique [A]. International Conference on Computational Intelligence and Security [C]. IEEE, 2015: 412-415
2. [The prediction model of online consumer reviews performance based on sentiment mining and Tobit regression](https://www.engineeringvillage.com/search/doc/abstract.url?&pageType=quickSearch&usageZone=resultslist&usageOrigin=searchresults&searchtype=Quick&SEARCHID=0ac90fc3M1245M41b9M9a97M8ce4fbe680a1&DOCINDEX=1&ignore_docid=cpx_M3ef4e6e015aa49a11f6M714510178163176&database=1&format=quickSearchAbstractFormat&tagscope=&displayPagination=yes)

# 致 谢

时光飞逝，一转眼研究生生活即将结束。回首两年半的学习时光，内心感慨颇多，也向曾经鼓励、帮助过我的老师、同学、朋友、亲人致以最诚挚的谢意。

首先，衷心感谢我的研究生导师曹庆华教授。曹老师治学严谨，学识渊博，不仅在学习和工作中都给予我悉心的指导和帮助，同时在生活中给予我指引和教诲，让我更加明确人生方向。同时曹老师极大地工作热情和尽职尽责的工作态度深深的影响着我，让我在今后的成长道路上时刻保持永不松懈的精神，不断以更高标准严格要求自己。在论文研究阶段和实验阶段，曹老师都给予了非常多且重要的指导，让我能够沿着正确的方向研究下去。此外，曹老师不仅在学习生活上给予我无微不至的关怀，使我顺利完成学业，还教会我很多做人做事做学问的道理。这是我攻读硕士学位期间最大的收获和一生中最宝贵的财富。祝曹老师工作顺利，桃李满天下。

衷心的感谢孙青老师在两年半的时光中对我学业和生活的关心和支持。在研究内容尚未清晰时，孙老师耐心地与我探讨相关工作，为我开拓思路，并尽可能地帮助我寻找解决问题的方法。在投递小论文时，孙老师非常耐心细致地帮助修正语法错误和不正确的内容，对我的帮助特别大，在此感谢孙老师的无私的帮助。同时，孙老师解决问题时的钻研精神和一丝不苟的工作态度非常值得我学习。在此祝愿孙老师在未来的工作和生活中一切顺利，幸福美满。

感谢实验室的刘云师兄、庞静雯师姐在我学习和生活中遇到困难和挫折时对我的鼓励和帮助；感谢王鑫冶、田庆松、苑铎同学在实验室时常与我讨论问题，开拓了我完成论文时的思路，同时感谢同学们在找工作时对我的帮助。衷心祝愿大家前程似锦。

感谢我的父母，感谢你们对我一直以来的支持，感谢你们在我求学生涯过程中对我无微不至的关心和鼓励。无论顺境逆境，只要想到你们会一直站在我的身边，都会充满力量，继续走下去。祝你们身体健康，永远爱你们。

感谢本文引用文献的众多作者们，你们的文章开拓的我解决问题的思路，也为论文提供了理论依据和基础。

最后，感谢审阅本论文的评审老师和答辩组老师们，感谢你们对论文的指导和宝贵的修改意见。