# 绪论

本章将首先介绍论文的研究背景和研究意义；其次对目前国内外关于软件需求跟踪任务所使用的方法和软件工程领域文本检索算法进行了总结和分析，指出了每种方法的应用场景和各自的局限性；然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《航空发动机电子控制系统适航审定关键技术》子项目《航空发动机控制系统软件和电子硬件适航审定方法与流程研究》。

2017年商飞C919客机的首次试飞顺利完成，标志着我国的民用航空事业进入了新的历史阶段。民用航空事业的基本理念是“民机发展，适航先行”[1]。所谓“适航（适航性）”是指航空器的各个部件和子系统能够在预期的环境下安全起飞、着陆和飞行的固有品质，并且这种品质能够由航空器全寿命周期的各个环节来完成和保持[1]。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）在软硬件的功能分配上，研制人员更青睐于使用软件。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，因此如何保证在整个生命周期过程中的软件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标准的基础上提出DO-178C，即《机载系统和设备合格审定中的软件考虑》。该标准面向过程和目标，明确了研制满足适航性的软件需要实现的过程和需要达到的目标，但是并没有严格约束开发流程，以可协调的方式为开发适航软件提供了指南。DO-178B/C在标准层面上制定了适航软件开发的流程以及应该达到的目标，其中制定的目标中明确指出：1）高级需求可追踪到系统需求；2）低级需求可追踪到高级需求；3）源代码可追踪到低级需求[2]。

需求跟踪，即通过建立需求同软件生命周期中各个阶段元素之间的联系，实现对每个需求的监督和跟踪。在工业实践中，现有的需求跟踪技术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增多，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，这些软件在记录软件需求跟踪关系的同时，能够帮助软件相关人员增强对项目目标的沟通，增强协作开发的效率。同样的，这些软件也是通过建立需求跟踪矩阵来实现，尽管使用这些软件能够给信息录入人员简化部分操作，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，自动化的需求跟踪技术主要基于信息检索思想，使用向量空间模型（Vector Space Model，VSM）、潜在语义索引（Latent Semantic Indexing，LSI）和概率模型（Probabilistic Model，PM）等模型[3]。这些方法基本是通过将软件文本表示成词频向量的形式，然后通过向量计算文本相似度，很少考虑软件文档的语义。近年来越来越热门的自然语言处理（Natural Language Processing，NLP）能够通过统计学和语言学的理论计算得到词和词、句子和句子之间的语义关系，从而提升文档关系计算的准确率。同时，随着机器学习技术的发展，更多研究人员利用文本和文本间的特征对文本分类，并且可以通过训练的模型对未知文本进行预测。因此，在软件需求跟踪的过程中加入语义计算和机器学习算法，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，同时自动化的需求跟踪技术的研究更是对规模日益增长的适航软件系统的研制和审定具有重要意义。

## 国内外研究现状

### 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，其中静态需求跟踪技术主要使用需求跟踪矩阵、需求跟踪图和交叉引用等多种静态文本形式表示跟踪过程，动态需求跟踪技术，即上文提到的自动化建立跟踪关系的技术，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、基于一定的增强策略和基于规则的方法[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，但随着软件规模的增长，软件生命周期越来越长，静态跟踪技术易于出错、不易维护的缺点就暴露出来，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）用来表示需求和生命周期过程中各个元素的联系，包括与设计阶段、实现阶段和测试阶段等阶段元素的联系。设计阶段元素包括系统设计图、数据流图、类图、数据库ER图和设计文档等，实现阶段元素包括源代码文件、类描述文件等，测试阶段元素包括测试用例文档和测试结果报告等。需求跟踪链接和软件生命周期中其他过程的元素可以具有多种关系：一对一、一对多和多对多。举例说明，一个需求可以和一个或多个代码文件相关，同时一个代码文件也可以对应一个或多个需求。需求跟踪矩阵示例如图1所示。



图1 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、需求和各个阶段元素之间的关系，以及在需求变更时将受到影响的各部分元素。需求跟踪图对跟踪关系的展示比较明显，同时既能够表示需求和系统中各个元素的关系，也能够表示开发过程中产生的中间元素之间的关系。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。

交叉引用用于建立软件需求文档之间的跟踪关系，需求文档包括需求规格文档、需求说明文档等，对于适航软件，还有系统需求文档、高级需求文档和低级需求文档等。使用交叉引用建立的跟踪关系较直观，方便实际使用，但只适用于需求文档之间跟踪关系的建立[5]。

（2）动态需求跟踪技术

动态需求跟踪为解决手工建立需求跟踪关系易于出错和难以维护的问题，使用自动化或半自动化的技术建立和维护跟踪关系[4]。

由于软件在全生命周期的设计、开发和维护过程中产生了大量的文本信息，因此动态需求跟踪技术大多从文本角度出发，将需求文档看作信息检索任务中的待查询文档，并计算其与不同软件制品之间的相似度，设置阈值，相似度高于该阈值的软件制品则被认为与检索的文档具有跟踪关系[6]。

文献[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。该方法的核心步骤是使用语义聚类对得到的不同分组进行映射，然后使用得到的结果来建立跟踪关系。

文献[7]和文献[4]是同一个作者，文献[7]通过命名实体识别来恢复软件需求和源代码之间的跟踪关系。该文章作者表示基于文本的需求跟踪方法的准确率会随着文本质量的高低而有所不同[7]，因此，该文章提出使用代码上下文建立命名实体识别模型，使用命名实体表示文本中的关键词，解决抽象语法树无法解析自然语言文本的问题。文中使用基于最大熵模型的需求跟踪方法，将源代码中的命名实体识别看作分类问题，同时将代码的类别分为：class、method、invocate、normalText、comment、param；然后对上述已经找到命名实体的制品文本进行预处理，将软件制品转为文档集合，然后使用文献[4]中相同的聚类方法对关键词聚类和映射，根据映射结果建立跟踪关系。

由于之前的工作没有充分利用过代码文件中的代码注释，因此文献[6]中提出使用代码注释来提升需求跟踪任务的准确率，在需求文档和代码之间建立更精确的跟踪关系。文中指出动态需求跟踪任务使用信息检索技术，建立需求文档和其他软件制品的跟踪关系，在跟踪关系的精度上不能满足要求，同时文中认为问题一部分是出在没有利用注释信息上面。该文章中使用的方法和文献[7]相似，主要有两点不同：1）数据预处理阶段，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）使用向量空间模型表示文本需求和源代码文件，然后使用向量空间模型中的向量计算二者的相似度，设置阈值，达到该阈值即可建立跟踪关系。

以上三个文献分别利用句法分析、命名实体识别和代码注释作为增强策略，从各自的角度和原始的的信息检索方法进行对比，效果都得到一定程度的提升，但是整体效果都比较弱。

文献[8]是比较早被提出通过信息检索的方式来完成动态需求跟踪的文章。文中提出解决需求跟踪任务的主要步骤包括：1）将任务构造成信息检索问题；2）选择IR算法；3）将需求文本输入到IR算法中；4）分析算法输出；5）选择合适的策略整理算法输出；6）比较该方法与其他方法的性能。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）高层需求和低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。该论文提供了动态需求跟踪链接的基本思路，也给了我们在该方面研究的启发。

文献[9]提出基于使用LSI，即潜在语义索引，建立需求和设计制品、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，将所有的文档都表示为LSI的子空间，通过余弦相似度方法计算文档间的关系。同时文中指出LSI不依赖于提前定义的单词表或语法，也就是说，可以省去部分耗时耗力的数据预处理过程。但是该方法存在着LSI方法自身的不足之处，该问题将在2.2.1节详细描述。

### 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[10,11]、特征定位[12]、软件复用[12]等。为了提升文本检索的性能，很多方法被提出，下面将列出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扩展待查询的单词或短语，去解决“词汇问题”[13]。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[14]指出大部分的自动查询扩展都显式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，赋予不同的扩展词不同的重要程度，保证扩展后的单词集合或短语集合能更完整的表达待扩展词的含义[15]。

然后是词向量（本文中词向量特指Word Embedding），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，其中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，而skip-gram模型与CBOW的输入输出相反。

在软件工程任务中，很多文本检索任务都使用了word2vec或word2vec的改进模型。X. Ye等人[16]提出一种新的学习词向量的方式，并使用训练得到的词向量去计算文档之间的相似度。在该文章中，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“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[18]结合了word2vec和循环神经网络（Recurrent Neural Networks，RNN）去建立软件制品之间的跟踪关系，通过RNN预测两个制品的语义相似度。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。同时，事实上，得到包含有足够跟踪链接的大规模工业软件数据也是比较困难的。

另一个和文献[18]相似的工作是使用机器学习分类器去估计软件文档中跟踪链接的数量[19]。文献[19]中作者使用了多种不同的分类算法和不同的自然语言处理方法进行两两组合去探寻最预测模型的最优精度，该文献还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，去恢复软件需求跟踪链接。

Tien-Duy B. Le等人[20]使用文本检索和程序频谱（Program spectrum）解决故障定位问题，文中使用三种方法计算了三种相似度，分别是：使用向量空间模型计算得到的相似度、使用论文中提出的方法Tarantula计算得到的相似度和使用基于可疑词的方法计算得到的相似度，然后将三者通过三个参数结合起来，作为最终的相似度。Shaowei Wang等人[21]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，都是根据给定的文本或者代码文件去查找与之相关的文本或代码文件。但是，故障定位任务往往使用一些和程序故障相关的特征去构建相关模型，比如使用程序频谱，即程序执行成功或者失败的路径，这些特征在需求跟踪任务中无法获得，因此该类任务的方法难以直接应用到软件需求跟踪任务中。

另一个相关的工作是文献[22]，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。

## 研究目标和研究内容

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辅助依据。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之间的关系如图2所示。



图2 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，明确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，和在软件设计、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、学习排序算法模型的改进以及对需求跟踪任务精度的提升。

（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ENRL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，并对IR SVM学习排序算法损失函数中的权重进行改进，然后使用改进的学习排序算法对需求跟踪链接进行进一步的优化，最后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。

## 论文组织结构

本文共分六章，各章节编排以及主要内容如图3所示：



图3 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和软件工程领域文本检索算法的国内外研究现状，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，以及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理论依据和研究方向。

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 based on Word Embedding 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改进了文本语义相似度计算算法，接着使用已经计算好的文本语义相似度改进了IR SVM算法损失函数中的权重设置方式，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。本章设置了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原型系统构建。本章介绍了原型系统的需求、开发环境和系统的总体设计，然后介绍了原型系统中的主要功能模块的设计与实现，同时展示了该原型系统的主要功能。

总结与展望。总结本论文的研究工作，并提出尚未解决的问题，制定今后的研究工作重点和对该领域发展的展望。