# 相关理论与技术研究

本章主要介绍适航领域相关理论、软件需求跟踪领域理论和相关技术。首先介绍了适航领域中机载软件相关的标准DO-178B/C，详细说明了与需求跟踪相关的适航审定目标；然后对信息检索技术的常用技术和模型进行介绍；接下来介绍了本论文中使用的word embedding；最后介绍了学习排序算法的基本概念，以及在软件工程领域中的使用情况和已有的方法。

## 适航领域相关理论

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世纪70年代末，计算机软件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制定了相应软件规则，用来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、软件开发以及软件综合过程，三个过程结构图如图4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、开发、验证、构型管理以及质量保证等计划，三个标准包括软件需求、设计以及编码等标准。与开发和设计人员关系最密切的软件开发过程由软件需求、设计、编码和集成等四个过程组成。软件综合过程包括软件验证、构型管理、质量保证以及审定联络等四个子过程，并且综合过程在软件全生命周期中起作用。

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同时执行，并且软件综合过程执行在软件开发过程的各个子过程上，具体的关系如图5所示。虽然标准中对软件生命周期过程规定较多，但是标准并没有强制所有软件活动必须按照规定活动严格执行，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[2]。



图4 DO-178B/C生命周期过程结构图



图5 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中列举了20种软件生命周期数据，其中包括软件计划过程中的五个计划数据和三个标准数据，软件开发过程中的软件需求数据、设计描述数据、源代码数据、可执行目标代码数据和软件验证用例和规程，以及软件综合过程中的软件验证结果、环境构型索引、软件构型索引、软件构型管理记录、问题报告、质量保证记录以及软件完结总数。其中本文将使用到的数据包括软件需求数据和源代码数据。

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6所示。



图6 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，可能会产生衍生的高级需求[1]，即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直接从系统需求追溯过来，因此需要提供一些衍生的高级需求来完成更具体的功能；同理，有些高级需求描述的功能，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。

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低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7展示了以上的可追溯性关系。

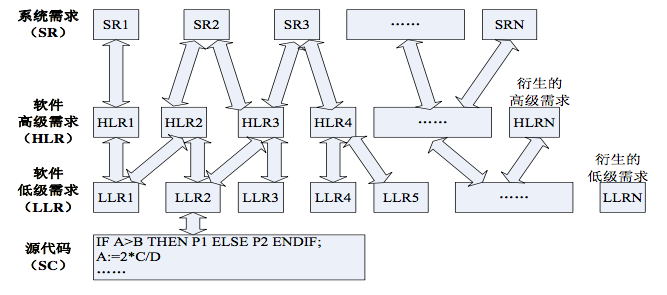


图7 DO-178B/C中要求的追溯性关系

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[23]，最后将结果按照相关度排列。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结果来确定查询结果。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的内容。当前常用的基于统计检索的方法包括向量空间模型和概率模型。

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，从而将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[24]。

潜在语义索引也是基于向量空间的检索方法。LSI使用one-hot方法表示文档，并将所有文档放在同一个矩阵中，通过奇异值分解（Singular Value Decomposition，SVD）的方法对矩阵降维，并且得到文本的主题和主题之间的关系。但是由于使用向量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。LSI方法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。

概率模型基于多个随机变量之间概率关系计算查询项和候选文档之间的关系，将查询项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(2.1)所示。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，从而完成信息检索任务。

（2.1）

其中表示文档集合中第*i*个文档，表示查询语句，公式(2.1)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通过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，如在文献[25]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[26]和基于外部语义词典（如Hownet[27]和Wordnet[28]）等方法，这些方法基本是将查询文本中的单词或词组抽离出来，然后用本体或者外部语义词典中的含义来表示，从而将整个文档集合中的内容统一表示，然后使用给定的算法计算相似度，从而确定检索列表。

## word embedding介绍

Word2vec是Google的Tomas Mikolov[29]等人提出的文本表示方法，将单词表示成word embedding的形式，是一种分布式表示模型。Word embedding的向量表示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成向量的形式，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[30]。训练word embedding的流程类似前馈神经网络，但是仅包含输入层、隐藏层和输出层三层结构，并且由于Mikolov等人对隐藏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skip-gram和CBOW两种，skip-gram模型的输入是特定词的词向量，输出是特定词上下文相关的词的词向量，而CBOW和skip-gram的输入输出相反。skip-gram和CBOW模型分别如图8和图9所示。



图8 skip-gram模型示意图



图9 CBOW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[31]，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和数据重复检测等方面。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10所示，更加形式化的表示如公式(2.2)和公式(2.3)所示。

(2.2)

(2.3)

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，该问题转化为分类或回归问题，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。该方法通过考虑两两文档对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所对应的所有搜索结果的列表，该方法直接训练优化算法模型输出的整体序列，因此其结果能够更接近真实的文档序列。已有的且常用排序算法如表1所示。



图10 学习排序的流程

除了排序算法，选择合适的特征作为模型的输入对学习排序算法的性能也有很大的影响。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，比如文献[32]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，文献[33]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[33]，其中依赖查询的特征即与查询文本相关的特征，如相似度特征；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[33]、故障修复频率[32]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1 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[34] 分类：SVM、McRank[35] |
| 文档对方法 | IR SVM[36]、Ranking SVM[37]、RankBoost[38] |
| 文档列表方法 | ListNet[39]、AdaRank[40]、SVM Map[41] |

## 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领域中常用的技术，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的概念和两种训练模型的区别；最后对软件工程领域的学习排序算法的应用以及常用排序算法进行介绍。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。