# 第二章 相关理论与技术研究

## 2.1 适航标准研究

### 2.1.1 适航认证体系

多个国家和组织通过建立适航认证体系来保证航空器的适航性，航空器在投入使用前要通过相关地区的适航审定标准，主要的相关标准有：美国联邦航空管理局的联邦航空条例 (FAR)、欧洲航空安全委员会的联合航空规则(JAR)及中国民航总局的中国民用航空规章(CCAR) [26]等。

目前国际上比较权威的适航规章体系是美国的联邦适航规章以及欧洲适航认证标准，美国的适航审定体系分为四个层次：“适航法规”、“咨询通告”、“标准”以及“应用”。对其解释和内容举例见表1。

表1 适航审定体系

|  |  |  |
| --- | --- | --- |
| 文件类型 | 文件解释 | 与发动机控制软件有关的内容举例 |
| 法规（FAR） | 必须遵守，强制执行 | FAR25部：运输类飞机适航规定 |
| 咨询通告（AC） | 认证机构对法规的理解及提出使用法 | AC20-115B:推荐RTCA DO-178B  AC20-115C:推荐RTCA DO-178C |
| 标准（ARP series） | 实施时过程、活动应该达到的要求 | SAE ARP 4754A：民用飞机和系统开发指南  DO-178B/C：机载系统和设备合格审定中的软件考虑  SAE ARP 4761：民用飞机机载系统和设备安全性评估过程的指南和方法 |
| 实践 | 选择特定的方法，采用相应的工具，实现一系列的活动，满足标准，符合咨询通告，遵守法规 | FAA　Job Aid及补充文档：软件审查工作指南 |

由表1可以看到，在咨询通告AC-115B/C[27,28]中，就机载软件的符合性证明中分别推荐使用RCTA DO-178B/C，我国整个适航体系架构和内容都和美国联邦航空局(FAA)基本相同。DO-178B/C是整个民航适航认证体系的一个非常重要的标准，一直以来，DO178B/C都在国内外民用飞机机载软件系统的研制过程中被采用，对于航空发动机控制软件系统的适航审定也是如此。

### 2.1.2 DO-178B/C标准研究

20世纪70年代末期，随着数字计算机被越来越多的应用到飞机设备和系统中，美国航空无线电技术委员会制定了相应的软件规则，用来支持指导管理当局对机载软件系统的适航审定。自此之后，《机载系统和热备合格审定中的考虑》即DO-178系列诞生，经历了DO-178、DO-178A、DO-178B和 DO-178C等阶段[3]，通过与实践结合，不断检验和完善。其中，DO-178B在其使用的近20年里没有发现严重问题，经受住了时间和实践的考验。随着软件研发技术的不断发展，如今的技术水平已经与20年前有了较大的变化，这些新技术不能完全满足面向过程和目标的DO-178B的要求。因此，FAA和EASA在2005年共同授权小组对DO-178B内容进行修订，并于2011年底通过审核并正式颁布，即DO-178C。DO-178C是对DO-178B标准的继承和扩充，并没有实质上增加机制软件适航符合性的要求。新补充条款包括：DO-330软件工具验证考虑、DO-331基于模型驱动的开发和验证、DO-332面向对象和相关技术、DO-333形式化方法[29-32]等。



图 5 DO-178B/C软件过程间关系

软件生命周期是指软件产品从概念提出到产品“死亡”的整个周期。就机载软件而言，其生命周期根据DO-178B/C标准可分为软件计划，软件开发及软件综合三个过程。其中软件计划过程主要起到对其软件开发和综合过程的活动的定义和协调。而软件开发过程与开发和设计人员关系最密切，它包括软件需求、设计、编码和集成等四个过程。软件综合过程则用于保证软件生的命周期及软件输出正确，受控和可信的过程。它包括软件验证、构型管理、质量保证以及审定联络等四个子过程[3]，三个过程间的关系如图6所示。

在满足DO-178B/C标准的软件开发过程中，软件计划过程制定软件生命周期活动，是所有过程的起点，软件开发过程则是软件生命周期的主线，软件综合过程与软件开发过程同时执行，并且软件综合过程与软件开发过程的各个子过程上交叉执行。DO-178B/C软件生命周期方阵图6所示。虽然标准中对软件生命周期过程规定较多，但是并没有强制所有软件活动必须按照规定活动严格执行，只需要保证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[4]。



图 6 DO-178B/C软件生命周期方阵图

在整个软件生命周期中会产生的大量软件制品（Software Artifacts），即软件生命周期数据，它们是软件产品的表现形式，切实记录了软件的整个生命过程，局方以此基本依据对软件进行适航审定。在DO-178B/C中，列举了软件计划、软件开发和软件综合过程中的对应的多种软件生命周期数据。本文的目标主要是构建软件需求数据，设计描述数据，验证用例和源代码等数据间的追踪关系。

DO-178B/C的主要关注软件失效对于安全性的影响，例如，每个组件失效有多严重以及对于整个系统的安全性意味着什么。这种影响被分为五类，即灾难性的，危害性的，严重的，不严重的，以及没有影响的。分别记为Level A、B、C、D、E五个等级，其中Level A的认证是最严格，它需要提交大量的软件生命周期数据文档以软件满足DO-178C标准所要求71个目标的证明。之后等级所需满足目标数目逐渐减少，Level E表示对飞行的安全性没有影响，DO-178C的需求对它不适用。DO-178B/C作为目前航空领域和适航审定当局普遍认可的一种适航符合性审查标准，一经采用，则必须满足该标准对该等级软件要求实现的所有目标。航空发动机控制系统所对应的级别为最高等级Level A，也就是必须实现DO-178C所要求的71个目标。



图 7 数据、过程和目标间关系

### 2.1.3 DO-178B/C标准中规定的可追溯性目标

从图7可以看出，我们通过使用软件过程中生成的软件生命周期数据来证明其是否满足适航审定目标。DO-178B/C标准规定，在软件的生命周期中必须实现需求可追溯性目标。可追溯性是软件和系统在开发和维护过程中的重要因素，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要求至少完成以下几个可追溯性目标[3]：

1. 系统需求和高层需求之间的可追溯性

系统需求指的是机载系统在开发的过程中分配给软件的要求，通常是机载系统对于某个软件功能的描述；高层需求是软件某个功能的描述，用于满足功能、接口、性能和安全相关需求，它是若干个需求的集合，通常在软件需求规格说明书中描述。软件高层需求的正确性和完整性在软件的研制有重要的作用。需要注意的是，在需求细化的过程中还会产生衍生需求，这些衍生需求无法追溯到上层需求。

系统需求与高层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层需求。

1. 高层需求和低层需求之间的可追溯性

高层需求是描述了软件实现的某个功能，是若干需求的组合，需要进一步细化为可由编码实现的若干低层的需求和衍生的低层需求。低层需求在软件设计文档中描述，它可用于编码实现，它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。

高层需求和低层需求之间的追溯性是为了保证高层需求及衍生的高层需求全部被转化成了低层需求。

1. 低层需求和源代码之间的可追溯性

对于已经转化为源代码的低层需求，标准同样要求保持两者之间的可追溯性，主要是为了方便代码的复核和检验。

DO178B/C标准所要求可追溯性关系如图8所示。



图 8 DO-178B/C中的可追溯性关系

除此之外，DO-178C还在第11章显示的要求建立测试用例和测试规程，测试规程和测试结果间的可追踪性。这些追溯性关系数据需要提供给适航审定机构，它们被称为可追踪性数据（Trace Data）。这些数据直接的追踪示意图如图9所示。可以看到这些软件数据之间存在明显的偏序关系。在机载软件研制的实践过程中，软件生命周期数据通常是条目化的，它是最小粒度且具有独立语义的数据，我们可以称它为数据元(Data Element)。由于在软件研发流程的定义中，并不能确定各种数据元（如系统需求）的条数，它们随着项目活动的进行才会被确定。软件需求数据的可追踪性应该建立在数据元的粒度之上。使用粒度得当的数据元对数据进行追踪对软件验证活动有十分重要的作用。



图 9 数据追踪示意图

## 2.2 文本相似度技术

文本相似度是指通过量化的方法计算两个词语或者文本间的相似程度，广泛应用在信息检索、文本分类和智能问答等领域。文本相似度计算是一项复杂的任务，目前有很多学者它进行了研究。它的计算主要分为两个部分：一是适当的对文本进行表示，二是选择适当的相似度计算方法。

### 2.2.1 向量空间模型

向量空间模型（VSM）在上世纪在上世纪60年代由Salton等人[33]提出，主要思想是将文档映射到向量空间，从而向量化文档。随着计算机技术的不断发展，VSM算法被广泛应用于信息检索领和文本分类域。VSM算法的一个特点是只关心文档中词或短语出现的次数，即词频有关，而与该词条在该文档中出现的位置和顺序无关。VSM算法把的核心在于将文档数据的处理简化为向量空间中的向量运算，从而快速计算出文档间的相似度。使用VSM算法将文本数据转换成向量之后，可以通过计算各向量之间的相似性来度量各文本之间的相关程度。

VSM算法的基本原理可表示为：在VSM中，每一篇文档都可以表示成公式(2.1)的形式。

 (2.1)

在这里，表示文档第个特征词条，则表示第个特征词所对应的权重。

可以看出，如何计算特征词的权重的计算是VSM算法的核心内容。在计算文档中每个特征权重时，应当考虑文档中词语代表文档特征信息的能力，能力越强，应赋予的权重值越大。特征项的权重计算方法[34]有词频算法和TF-IDF算法。常用的是TF-IDF算法。

词频（Term Frequency, TF）算法，顾名思义，即用文本特征词在文档中出现的次数作为该特征词的权重。显然文档中出现次数最多的词具有最大的词频，即拥有最大的权重值。这并没有很好的结合文本特征，文本中的很多高频词汇并不能很好区分文档，如常用的形容词，名词，代词等，它们出现频率很多，但却不能很好的区分文本类型，相反有些低频词对文本的区分度却更强。词频算法只考虑到高频词语忽视低频词语在文本检索，分类中并不实用。所以在实际的应用中单独使用词频算法构建权重的实例并不多。可用公式(2.2)计算得到。

 (2.2)

其中，表示第个词汇在第个文本中出现的次数。

通过对只考虑词频构造特征权重的词频算法进行改进，在构建特征权重时，用TF-IDF（Term Frequency-Inverse Document Frequency，TF-IDF）算法用来衡量一个特征项对一个文档的重要程度，其权重计算方法是将文本特征项的词频和逆文档频率相乘所得的结果作为文档特征项的权重。词频（TF）认为在一篇文档中出现次数越多的词或短语代表这篇文档的能力越强。逆文档频率（IDF）则认为如果包含某个词或短语的文档越少，则表明该词条代表这篇文档的能力越强，就越重要。TF-IDF将两者的思想结合起来，某个词或短语在某一篇文档中出现的次数越多，并且在其他文档中出现次数较少，TF-IDF则认为该词或短语具有很高的类别区分能力。TF-IDF的计算公式如(2.3)：

 (2.3)

其中，代表在第篇文档中第个特征词出现的次数，代表训练语料库中文档的总数， 则训练语料库中含有特征词的文档数。

TF-IDF算法的思想并不复杂，它充分结合词频和文档逆频特征，认为在所有文档中出现频数高的词而在具体一篇文档中出现频数低的词条重要，在一定程度上抑制了高频词的作用。在实际的应用中，TF-IDF也较TF算法表现出了更好的性能。

用VSM算法表示文档时，计算两篇文档 和 的相似度有以下三种计算方法：欧式距离、向量夹角余弦和向量内积。

首先是欧式距离，可用公式(2.4)表示：

 (2.4)

其中，代表文档特征向量的维数或者是文档的特征个数，即不同单词的数量，代表文档中第个特征的权重，代表文档中第个特征的权重。

使用欧式距离度量文本间相似度时，欧式距离越小，即两文本之间相距越近则它们越相似。反之，若它们的距离越大，则它们的相似度就越小。

然后是用向量夹角余弦(cosine)计算相似度，可用公式(2.5)表示：

 (2.5)

其中，代表文档特征向量的维数或者是文档的特征个数，代表文档中第个特征的权重，代表文档中第个特征的权重。

使用向量夹角余弦度量文本相似度时，余弦夹角值越小，则文本间的夹角越大，它们的相似度就越小。反之，当夹角余弦越大时，则它们的相似度就越大。

最后介绍使用向量内积来计算相似度，可用公式(2.6)表示：

 (2.6)

其中，代表文档特征向量的维数或者是文档的特征个数，代表文档中第个特征的权重，代表文档中第个特征的权重。

使用向量内积度量文本间相似度时，若两个文档特征向量的内积越大则它们的相似度越大；反之，它们的相似度越低。

VSM的一个特点是不考虑文档词或短语的位置与顺序关系，这极大简化了在计算文档特征权重时的计算任务，但也失去了许多关于文档结构和语义的相关信息。在实际应用中，向量空间模型是目前使用最广泛的文档表示模型之一。

### 2.2.2 潜在主题模型

潜在主题模型(Latent Topic Model)基于具有相似含义的词汇总是出现在相似的语境中的假设，利用文本中具有相似语境的词汇间的联系，将文本空间映射到具有更低维度的向量空间中。对于语义关系相近的词来说，它们的向量表示在向量空间中距离较接近，可以说它们拥有相同的主题，且可以用余弦距离来衡量它们的相似度。

LSI是主题模型的一种，首先，可以将文档集表示为一个稀疏矩阵，的列向量表示一个文档，矩阵的元素可以使词频或者TF-IDF表示。构建好矩阵后，LSI首先如公式(2.7)对矩阵进行奇异值分解(SVD)，其中为特征值矩阵。只保留其中个最大的特征值，将矩阵进行降维，公式(2.8)中的就是降维后得到的新矩阵。降维不仅能减小矩阵的规模，提高计算效率，还可以一定程度去除原文档中的噪声数据。最重要的是通过降维可以将文档矩阵中语义相关的部分合并，将原矩阵变换到一个潜在的低维主题空间。文档中的每个单词或者文档都可以用该空间的一组权值向量表示。这些权值反映了与对应潜在主题的关联程度的强弱。得到单词或文档的向量表示之后，就可以用余弦距离或者其他方法来衡量它们之间的相似度。

(2.7)

(2.8)

## 2.3 词嵌入简介

统计语言模型(Statistical Language Model，SLM)是表示语言基本单位(通常为句子)的概率分布函数，即该语言的生成模型，它被广泛的应用于自然语言处理领域的语，是目前语言模型的基础。为了更形象的说明，假设句子由个词顺序构成，即，则句子的概率可以由的联合概率表示为公式(2.9)：

(2.9)

利用Bayes公式，上式可以分解为各个词语条件概率的乘积，如公式(2.10)：

(2.10)

其中为词语的上下文，这些条件概率就是语言模型的参数，可以根据这些条件概率的值，计算出句子的概率。

Bengio等人[35]提出的神经概率语言模型(Neural Probabilistic Language Model，NPLM)，这种方法利用神经网络算法来构建语言模型，将词语嵌入固定维数的向量(Distributed representations)。



图 10 Skip-gram模型示意图



图 11 CBOW模型示意图

词向量的概念最早由Hinton[36]在1986年提出，是一种把词语映射到实数空间固定维数向量的技术，即词嵌入，这种方法的理论基础是具有相似上下文的词语具有相近的意思，因此，具有相似词语的向量在向量空间中距离很近，而这可以看作词语的语义的特征。

Thomas Mikolov等人[37,38]对传统的NPLM做了两个改进，一是将RCNN引入NPLM，使之可以处理变长序列，另一个是在输出层加入softmax函数，通过开启从而加快了训练速度，使这项技术被广泛应用于自然语言处理任务中。在文献[39]中，他还给出了两个模型，Skip-gram和CBOW，来帮助使用者更好的训练词向量，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9和图10所示。

## 2.4 学习排序

学习排序(Learning to Rank, LtR)是一类自动学习将信息检索系统返回的结果更好的进行排序的机器学习方法的统称[44]，它主要研究的问题是如何构建查询和候选文档之间相关程度的函数或模型。它在自然语言处理、信息检索和推荐等领域有重要的作用。在软件工程领域，主要用于故障定位，测试用例和代码间可追踪性等等问题中。

学习排序在信息检索中的所要解决的问题可以描述为：给定训练集和测试集。 中文档用三元组表示为，其中为查询文档和为候选文档集的组合，表示两者的相关性，相关的记为正相关，否则记为负相关。其中作为机器学习算法输入的和是从文档中提取的特征，通过排序算法得到可以用公式(2.11)表示。常用的学习排序特征有语义相似度，文档逆频率和等。

(2.11)

其中，函数将查询-文档对映射为特征向量，是表示特征向量的权重矩阵，是排序函数。

按照训练样本类型的不同，现有的学习排序方法可以分为三种类型单文档方法（Pointwise）、文档对方法（Pairwise）和文档列表方法（Listwise）。Pointwise方法的训练对象是单个文档，通过将文档转化为特征向量的方式，把排序问题转化为了分类和回归问题。Pairwise则是将任意一对文档(,)当作算法的输入，最后输出结果是文档， 的相对位置关系，即和哪一个应该排在前面，此时的排序问题被转化为了了二分类问题。Listwise方法的训练样本则是对应每个查询的候选文档列表，其将整个文档列表作为一个整体，在数据足够大的情况下，往往有最好的效果。

表2 常见的排序算法

|  |  |
| --- | --- |
| **排序方法分类** | **常见算法** |
| 单文档方法(Pointeise) | Discriminative model for IR、SVM、McRank |
| 文档对方法(Pairwise) | Ranking SVM、RankBoost、IR SVM、RankNet |
| 文档列表方法(Listwise) | RankCosine、ListNet、ListMLE、SVM-Map |

在学习排序中，排序特征的选取同样会对算法的结果产生较大影响。在软件工程中常使用有词语或者文本相似度、上下文相似度还有词频，文本长度等特征。排序学习中的特征根据是否与查询语句相关可以分为两类：依赖查询特征和不依赖查询特征[45]。其中相似度特征就属于查询依赖特征，文本长度则是不依赖查询特征，它与查询文本无关。合理的选择特征的种类和数量有助于提升结果的精度。

## 2.5 本章小结

本章首先对适航领域的相关内容进行了研究，包括目前国际适航审定的体系，针对软件适航审定的DO178B/C标准，及DO-178B/C中所规定的软件开发过程，软件生命周期数据和对于可追溯性的目标要求，然后介绍了一些在需求追踪领域常用的文本相似度算法，接着介绍了词嵌入的语言模型，包括基于统计的语言模型基础，及基于神经网络的模型，最后介绍了机器学习排序算法相关理论，这些是本文后续研究的理论和技术基础。