## 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。