为了能够对变更管理中的一部分涉及到具体内容的情况进行审定，本文提出利用文本相似度算法对需求引起的变更前后的相关配置项进行初步验证的方法。

该方法适用于对追溯性体现的较为明显的生命周期数据的变更进行验证审查，具体为 系统需求、高级需求、低级需求、代码等。

具体验证逻辑如下：

**1.变更前，判断变更影响分析的正确性。**

变更执行前，使用文本相似度计算的方法判断系统需求、高级需求、低级。。。间的相似度，并根据计算出的相似度，确定各个层次需求间的追溯关系，将该追溯关系与实际变更中被更改项、受影响项进行比较，初步判断变更影响的正确程度。若是提供了追溯矩阵，则可以用于对追溯矩阵进行检查。

**2.变更后，判断新生成配置项追溯性的正确性。**

变更执行后，计算被变更的需求与受变更影响新生成的配置项的相似度，得到新生成的配置项是否满足追溯性关系，来确定变更实施是否正确。

通过图+说明 表述清楚

下一章介绍利用词向量计算相似度的方法及改进。

下一章对相似度算法进行验证Wordvec是目前用于计算文本相似度的较为常用的方法，获得生命周期数据的追溯性的过程包括：一、训练词向量 二、获取易于处理的生命周期数据格式 三、将生命周期数据映射为词向量并计算相似度 四、根据设置的相似度阈值确定生命周期数据间的追溯关系。

对于变更前基线LinePre， P为全部系统需求的集合，HT为全部高级需求的集合，LT为全部低级需求的集合，对于一次由高级需求改变而引起的变更，经变更后，建立新的基线LineCe，新的系统需求集合为ST\_c,高级需求集合为HT\_c，低级需求集合为LT\_c。

该需求对应的配置项C1发生改变，该变更引起的其他配置项的集合为Q，经过变更后，在新的基线中，原配置项C1变更为配置项C1\_m,并生成了新的配置项集合Q\_m,

### 4.0 利用词向量计算相似度的方法

利用 词向量Wordvec是目前用于计算文本相似度的较为常用的方法，获得生命周期数据的追溯性的过程包括：一、训练词向量 二、获取易于处理的生命周期数据格式 三、将生命周期数据映射为词向量并计算相似度 四、根据设置的相似度阈值确定生命周期数据间的追溯关系。



对于语料库的获取，我们选取了维基百科数据集，并加以处理，获取专业性更强，适用于软件开发的用来训练词向量的语料库。并以此训练词向量作为文本相似度计算的向量使用。

然后通过预处理获取需要进行相似度计算的生命周期数据，并映射为词向量，输入到计算文本相似度算法中

然后，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，即把每个需要建立跟踪链接的软件需求文本看作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，排在最前边的是与相应查询相关度最高的软件制品。

下面主要详细介绍对文本相似度计算算法的改进：

## 四、文本语义相似度计算方法

本节基于TF-IDF方法权重计算与缺失词替代，提出一种提高文本相似度计算效果的方法。

### 4.1基于tf-idf策略计算权重的关键词相似度加成算法

#### 4.1.1基于TF-idf策略计算权重

在计算文件相似度时，是先计算词与词之间的相似度，再计算出词与句子的相似度，最后再计算文本与文本间的相似度。但是不同的词能够代表文档信息的程度是不同的，如果将所有的词一视同仁，就忽略了关键词或词组对文档相似度影响更明显这一情况。因此，要对文档中的词计算特征权重，通过计算权重选出更能代表文档信息的词或词组。此处使用TF-IDF算法找出权重足够大的词汇TF（Term Frequency）是词频，如公式（）所示，指的是目标词在文档中出现次数的频率，tf与目标词能够表示文档信息的程度成正比，；IDF（Inverse Document Frequency）是逆文档频率，如公式（）所示，IDF与含有目标单词或词组的文档数量有关，用来表示目标词在其他文档中出现的频率，IDF值越大，代表目标单词在其他文档中出现的频率越小，则目标词或词组能更大程度的表示文档信息；最后将求TF与idf的乘积，如公式（）所示，得到该目标词的tf-idf值。

(3.1)

(3.2)

其中指目标词在文档中出现的频率，指单词在文档中的词频，表示文档中目标词出现的次数。指目标词在文档集合中的逆文档频率，表示文档总数，表示出现过目标单词的文档数，为了防止出现过目标单词的文档为零而导致分母为零，将该文档数加一处理。最后取对数处理。

#### 4.1.2对关键信息词进行加权计算调整后相似度

使用tf-idf选出能够更大程度反映目标文档信息的关键词后，需要使用专门加权方法计算相似度，以合理使用关键词所携带信息。

当计算关键词与其他词的相似度时，若两词的相似度大于阈值，则进一步按照一定比例提高相似度，应满足 ，具体数值通过实验确定，若两词的相似度小于阈值，则按照一定比例降低相似度，应满足 ,具体数值通过实验确定。计算方法如公式（）所示。

代表经过加权计算后得到的词与词之间的相似度， 代表直接计算得出的词与词之间的相似度，为取值大于1的常数，的值决定了对相似度进行调整的幅度。当或时，应用此公式。

当时，不对相似度进行调整，直接使用作为词与词间最终相似度。

##### 4.1.2.1以下为选出tfidf权重较大的单次或词组集合的步骤（意思不变，改，防相似）

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(3.4)所示。

(3.4)

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[42]。对某个文档进行查询扩展，是对文档中的各个单词或词组进行扩展，步骤如图12所示，描述如下：

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

（2）对以上选择出的前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(3.5)所示。其中参数同样设置为经验值0.7。

(3.5)



图12 查询扩展流程图

### 4.2对基于缺失词的处理策略

对于缺失词，建立词汇表，他的相似度，等于“相关词汇”的相似度的平均值。但计算词汇间的相似度时，先确定是否为同一词。

原来的策略：1.不考虑该词，2.统一替代为某一向量 3.？？？

对停用词的相似度计算方法

本文提出一种，由人工对缺失词 使用相关词 组合 进行替代计算的方法，具体为将 在计算某缺失此 A 与 另一词 C的相似度时，该相似度等于 A的替代词组“” 中每个词与c的相似度的平均值，

这样既可将缺失词也考虑到计算中去

缺失在计算相似度时，首先判断另一词是否为与该缺失词为同样的词，若相同，怎相似度为一，若不相同，再按照以上计算方法进行计算。

若某词汇属于缺失词，但 不在idf词库，直接使用向量表示

若某词汇术语缺失词，同时在idf词库，表示改缺失词较为重要，

### 4.3改进的文本语义相似度算法

最后基于以上权重调整策略与缺失词处理方法，提出优化的文本相似度计算方法：

Ye X等人在[16]提出了用来计算文本与文本间相似度的算法W2V，该方法计算文本语义相似度的步骤为：

1）通过计算两个词向量的cosine相似度得到词与词之间相似度，如公式（）所示；

(3.4)

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。

2）计算单个词与文本之间的相似度，如公式 （）所示，计算目标词与文本中所有词的相似度，取最大值即为目标词w与文本T的相似度。

(3.6)

3）计算文本与文本间的相似度。如公式(3.7)所示，算出文本中的每个词与文本的相似度，并求和，最后进行归一化处理，即得到文本与的相似度。

(3.7)

(3.8)

其中表示文本中与文本的相似度不为0的词的集合。

（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(3.6)，如公式(3.9)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，遍历整个范围，当效果最好时，设定值，因此参数根据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[16]中相同的方法，即使用公式(3.6)计算。

(3.9)

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。

下面是对以上算法的改进部分

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

（2）改写计算词与词之间的相似度的计算方法，当两个词都不属于集合中时，算法不变，仍然按照公式（）计算，得到词与词之间的最终相似度，当两个词中有任一词属于集合时，计算出两词的相似度，并将

与设置好的阈值 、比较，

若，则

；

若或，则

其中设置为经验值0.3，设置为取经验值0.7，r设置为取经验值3。得出的作为最终的词与词间相似度，继续参与下一步的计算。

两词之间计算相似度的公式改写为公式（）。其中 a,b,c分别取经验值。。。。

对以上选择出的前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(3.5)所示。其中参数同样设置为经验值0.7。

(3.5)

（3）对原算法的第二步进行改进。在w2v算法中，算法得出的相似度与两个文本的输入顺序是相关的，两个文本在算法中并不完全对称，差异出现在计算词与文本间相似度时，不同的文本顺序会得出不同的结果；另外，当短文本是长文本的子集时，若拆分的文本是短文本，长文本中不属于短文本部分的词将会被忽略，计算出的相似度为1，与实际情况明显不符。为了保证计算文本相似度时计算结果的一致性，同时为了避免特殊情况下的错误，在新的算法中，作出如下规定：

计算两个文本间的相似度时，设较短的文本为，较长的文本为，则必须拆分长文本,通过计算中每个词与短文本的相似度，获得文本间的相似度。