# 模型验证

新的算法验证结构：

数据来源，来源于生命周期数据，进行预处理，取得基线的系统/高/低/代码等需求。

数据预处理

数据集选择

1. 实验设置 需求追溯的评估指标
2. 实验结果分析

# 适航领域软件需求跟踪算法模型验证

**本章通过实验对第四章提出的算法效果进行检验，并将该算法与其他经典算法对比，评估算法的性能。**

5.1实验设置：

5.1.1数据来源。

5.1.1.1 适航开发方提供的的生命周期数据  
 进行预处理方法

5.1.1.2 公开的可用数据集

5.1.2对比算法，

5.1.3指标介绍

5.2实验结果：

## 数据准备

取得适航开发中相关追溯性数据的方法：

1.来源于开发方，追溯性矩阵无误的需求数据及其链接。由于目前国内生命周期数据提供方提供的生命周期数据格式和类型一般无法达到Do178c规定的格式要求，所以为取得可用于实验的需求数据及其链接关系，需要专业人员将数据标注，并由机器或人工取得生命周期书记及其追溯矩阵，还需要领域专家对追溯矩阵进行检查，确定追溯矩阵正确性，获得正确性追溯矩阵与实验取得的追溯情况进行对比。并由专门人员将其录入数据库中。

2.以上是从已完成的适航项目中获取实验数据的方法，由于适航项目保密性强难以获得，特别是经过专家验证的追溯性准确的数据更难以获取，因此，我们选取了CoEST网站提供的几个公开数据集作为实验数据。CoEST是个为研究者进行软件和系统可追溯性研究提供数据集的网站，该网站提供了多个从实际的项目中整理而来的软件数据集，我们选择MODIS、CM1、eTOUR、EasyClinic和iTrust作为实验数据，这些数据集不但提供各层次的需求，而且提供了经过专业人员验证的追踪链接，能够显著减少我们进行数据标注以及人工录入、邀请专家进行链接标注与验证的工作量，同时由于这些数据集被研究者们广泛使用来评估算法的有效性，给我们对实验性能的对比提供了极大便利。

MODIS由NASA提供，包含其公开项目的部分需求数据，该数据集由Hayes[14]等人整理，被广泛应用在需求可追溯性的研究中；CM1来源于NASA基础项目的子项目CM1，它包含软件的高层需求和低层需求。eTOUR来源于一个面向旅游的导航软件，由Salerno大学开发； EasyClinic来源于一个医疗系统管理软件，同样由Salerno大学开发；iTrust来源于一个登记医用药品数据的开源项目。

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **公开数据集** | **高级需求** | **低级需求** | **用例** | **UML交互图** | **测试用例** | **类描述信息** | **代码类** | **追溯链接数** | **词条数** |
| CM1-NASA | 22 | 53 | × | × | × | × | × | 45 | 5767 |
| eTOUR | × | × | 58 | × | × | × | 116 | 308 | 97452 |
| MODIS | 19 | 49 |  | × | × | × | × | 41 | 4534 |
| EasyClinic | × | × | 30 | 20 | 63 | 47 | × | 1257 | 21882 |
| iTrust | × | × | 131 | × | × | × | 367 | 534 | 123501 |

图15 数据获取过程示意图

对数据进行预处理

需求数据经过整理后，根据类型可以分为两类：（1）文本文件（2）源代码文件。

在使用词向量相似算法对需求数据进行计算之前，由于在文本或代码中与文本内容无关的数字、词汇、符号等，为降低无关词汇对相似度计算的干扰。还需要对文本或代码进行以下处理：

1. 分词，通过空格，将文本或代码分割为单个单词。
2. 去除其中除了下划线和连字符之外所有的非数字、非字母符号，
3. 将所有单词转化为小写
4. 去除停用词。对于文本文件，去除的停用词时使用Python编写的NLTK[48]（自然语言工具包），NLTK 提供了大多数语言的停止词表；对于代码文件，去除的停用词是其代码语言的关键字

最后得到最终用于实验的需求数据文本。

这个图要重画



Siwei L.等人[45]指出在训练词向量时，语料的领域相关性比语料库的大小更重要，并且语料的领域性越强，词向量的表示效果越好。

获取了软件开发中产生的具有追溯关系的数据集后，还需要获取用来训练词向量的语料库，本实验中获取语料库的过程如图 （）所示，首先下载源数据，可选源数据包括百度百科、中文百科、维基百科等，其中维基百科提供数据库的下载，此处选择维基百科英文数据库作为源数据；第二，对源数据进行过滤处理，由于维基百科数据库特别庞大，数据库达到13.9G，同时，为了使获得的语料库领域性更强，对源数据进行过滤处理，此处选择《信息技术软件工程术语》作为词库，生成术语表，并使用术语表筛选维基百科源数据，得到用来训练词向量的语料库，最终生成的语料库大小为372M。

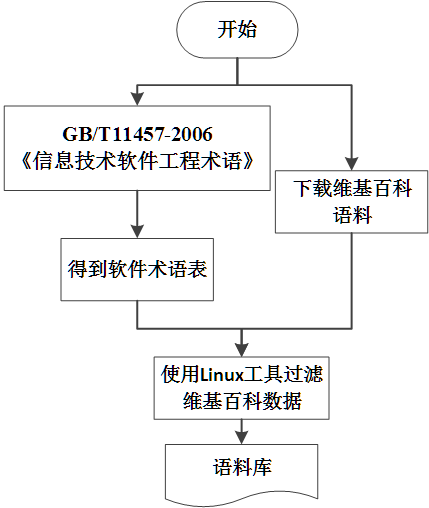


图16 获取训练词向量语料库流程

训练词向量：

本实验中使用开源项目Gensim来训练词向量，Gensim由python实现， Gensim是Radim Řehůře在其论文《**SCALABILITY OF SEMANTIC ANALYSIS IN NATURAL LANGUAGE PROCESSING**》中，使用python实现，集成了词向量模型，是一款很强大的工具。

在训练设置时，词向量训练模型选择Skip-gram，预测窗口大小设置为5，生成的向量的维度设置为200；同时选择softmax选项来提高训练速度。

## 实验设置

在实验中，使用本文提出的。。。相似度算法与LSI算法、W2V算法作对比，检验本文提出的改进的相似度算法的提升效果。LSI算法指的是使用潜在语义索引方法恢复软件跟踪链接的方法，该方法已经在跟踪恢复领域广泛使用，是一个比较成熟的方法。使用LSI方法进行文本相似度计算的流程如下：（I）将文本使用one-hot方法表示为矩阵（II）使用奇异值分解方法对矩阵降维到潜在语义空间；（III）通过对降维后的文本向量计算内积或者余弦相似度确定文本相似度。使用LSI作为对比方法是为了说明使用文本语义相似度计算比使用基于统计的检索方式效果更好。

表7 对比方法

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **实验** | **对比方法** | **文本词语顺序** | **基本方法** | **计算方法** | **文本/词向量 维数** | **算法改进** |
|  | LSI | 不考虑顺序 | 基于词频统计与文本向量空间 | 将文本表示为向量，并进行降维处理后计算文本相似度 | 通过SVD降维 | 无 |
|  | W2V | 考虑顺序 | 基于词向量空间 | 将词表示为向量，通过计算词相似度计算文本相似度 | 通过Word2vec确定词向量的维数 | 未使用IDF |
|  | 改进的文本相似度算法 | 考虑顺序 | 基于词向量空间 | 将词表示为向量，通过计算词相似度计算文本相似度 | 通过Word2vec确定词向量的维数 | 基于idf计算权重并对权重单词，对缺失词间相似度算法进行规定，进行加权处理 |

W2V方法是使用论文[16]中的方法来解决需求跟踪任务，算法核心在于对文本语义相似度的计算，如公式（3.7）所示。与W2V方法对比是为了通过实验验本文本相似度算法改进部分对性能的提升。

本论文中实验所用计算机使用八核英特尔i7处理器，内存为8G。

## 算法评估标准

为了精确、有效的对算法的性能进行比较。首先给出评估文本搜索领域算法性能的3个指标，分别是精确率、召回率和以及F测度。在描述模型性能评估指标之前，记：

为被检索文本列表中与查询的文本确实有追溯关系的文本数量。

为被检索文本列表中通过算法与被查询的文本建立了追溯关系的文本的数量。

为被检索文本列表中与查询的文本无追溯关系，却被算法错误的建立了追溯关系的文本数量。

为被检索文本列表中与查询的文本有追溯关系，却未通过算法得到追溯关系的文本数量

**精确率（Precision）：被检索的文本列表中通过算法确定了正确追溯关系的文本数量与通过算法建立了追溯关系的文本数量的比，可以被表示为**

**召回率（Recall）：被检索的文本列表中通过算法确定了正确追溯关系的文本数量与列表中确实存在追溯关系的文本数量之比**，可被表示为：

**F测度（F-measure）:**精确率和召回率越高，说明算法的效果越好，但是精确率和召回率之间是存在着相互影响的，为了提高精确率，召回率可能会降低；当召回率变高时，精确率可能会变低。F测度能够综合考虑精确率和召回率，用来表示精确率与召回率的调和平均值，F测度值越高，代表算法综合考虑精确率和召回率后的效果越好。

(4.3)

实验结果与分析

本节将详细描述实验结果，并使用前面提出的性能指标精对实验结果进行分析评估。

本实验分别使用LSI方法、W2V方法和WQI算法对数据进行相似度计算，根据计算出的相似度建立追溯性链接，并使用精确率、召回率和F测度对方法进行评估，实验结果见表（）。

表中，PRE代表方法结果的精确率，REC则代表方法结果的召回率，HReq、LReq、UC、CC、UMP、TC分别代表高级需求、低级需求、用例、类文件、UML交互图和测试用例。

比较表（）中展示的精确率和召回率，可以得出结论，基于词向量的W2V算法和WQI算法的效果在总体上要优于LSI方法。具体来说，与LSI方法相比，WQI算法的精确率相对高出了约33.3%，而召回率则高出了约24.5%；与W2V方法相比，WQI算法的精确率相对高出了约6.6%，而召回率则高出了约17.2%。

表8 三种方法的实验结果

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **数据集** | | **LSI** | | **W2V** | | **WQI** | |
| **PRE** | **REC** | **PRE** | **REC** | **PRE** | **REC** |
| CM1-NASA | HL→LL | 0.127 | 0.41 | 0.262 | 0.217 | 0.371 | 0.329 |
| GANTT | HL→LL | 0.286 | 0.332 | 0.278 | 0.418 | 0.255 | 0.563 |
| eTOUR | UC→CC | 0.077 | 0.221 | 0.098 | 0.332 | 0.088 | 0.415 |
| iTrust | UC→CC | 0.009 | 0.45 | 0.192 | 0.363 | 0.198 | 0.322 |
| EasyClinic | UC→ID | 0.259 | 0.833 | 0.338 | 0.75 | 0.342 | 0.806 |
| UC→TC | 0.45 | 0.755 | 0.522 | 0.867 | 0.499 | 0.867 |
| UC→CC | 0.317 | 0.503 | 0.215 | 0.677 | 0.232 | 0.76 |

实验使用的3种方法在五个数据集上，共有7组有追溯关系的数据，图（）展示的是这3种方法在这7组数据上的F测度，通过图（），我们可以直观地看出，XXX算法在6组数据上的效果都要优于LSI方法和W2V方法，对比W2V和发发发算法，发发发算法中对于IDF-df高的词的处理是合理的，能够使占权重高的词语更大程度代表两个互相比较的文本之间的关系，从而更清晰的区分出清晰度。

可以看WE、CWI算法在各个指标上对比LSI均有所提高。产生这种结果的一个主要原因是LSI算法在计算文本相似度时，使用奇异值分解的方法将文本映射到一个较低维的空间，具有相同主题的词语被映射到同一维度，而这种变幻是通过数学方法得到，并没有考虑词语的语义特性，而且LSI算法没有考虑词语之间的先后关系，相对于其他两个算法，忽略了词语之间的关系，因此在性能上要明显低于W2V和 。算法。

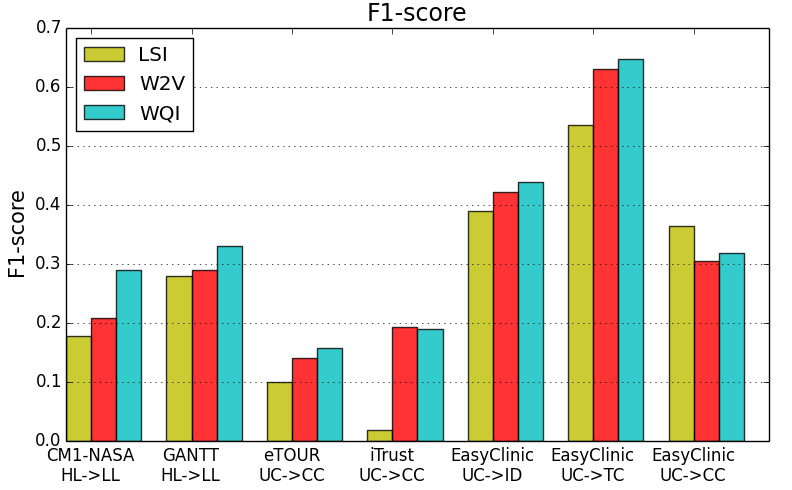


图18 LSI、W2V和WQI三种方法在七组数据上的F测度

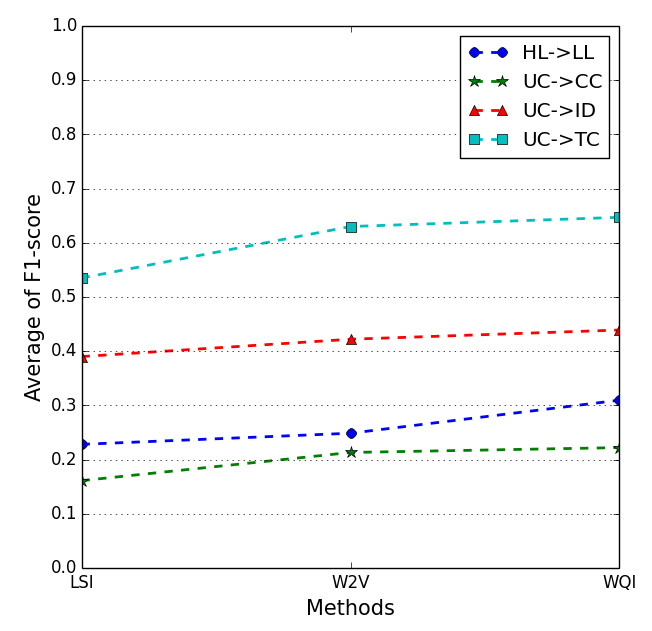


图19 LSI、W2V和WQI三种方法在不同类型的跟踪链接上的平均F测度

相比W2V方法，WQI算法引入了TFIDF加权策略和缺失词词向量处理办法，并且改进了计算的过程。从结果上来看，WQI算法所做的改进提升了结果的精确率和召回率，即提升了需求跟踪任务的性能。对于idf值较高的词，与文本含义的相关性更强，在算法中相对的提高这些词对文本相似度算法的影响，能够得到更好的效果；而对于缺失词利用相关词向量进行替代的方法，把缺失词的信息也有效利用，取得较好的效果。

综上所述，相比LSI方法和W2V方法，文本语义相似度计算算法WQI能够在软件需求跟踪任务中得到更好的效果。

本章小结

本章是本文的实验部分，目的是对本文所提出的软件需求跟踪模型Tr-WELR进行验证。首先介绍了数据准备流程和实验所用到的数据集的特征和来源，然后根据文本种类的不同分情况介绍了文档预处理过程，接下来对实验中的对比方法、训练词向量的语料库和实验环境做了介绍，并给出了模型的评估标准，最后对实验的结果和现象进行分析，给出结论。实验证明，在多组数据集和多个不同类型的跟踪链接上，本文提出的WELR模型能够较好的完成恢复跟踪链接的任务，）在软件文上使用提出的基于word embedding的文本语义相似度算法准确率有所提高；