# 适航领域软件需求跟踪算法模型验证

本章将通过实验验证和评估第三章所提出的模型Tr-WELR的有效性和性能。首先对验证模型所需数据集的来源和相关特征进行详细描述，然后介绍实验设置和实验具体内容，接下来对文档预处理流程进行详细介绍，然后介绍需求跟踪算法模型的评估指标，最后给出了详细的实验结果和分析。

## 数据准备

数据获取需要的步骤如图15所示，包括（1）相关人员对需求文档、设计文档、代码文件、测试文档、需求变更文档、审定文档和可能有的需求跟踪矩阵文档等项目相关文档的收集，对不存在需求跟踪矩阵或其他形式的需求跟踪文档的情况，还需要相关专家对部分需求的跟踪关系进行标注；（2）对数据进行脱密处理，常用的脱密方法如表5所示[44]；（3）对文档进行自动解析，根据文档的结构和标题，将不同生命周期的数据抽取出来，将半结构化数据转为结构化数据，然后录入人员对数据的分类情况进行确认；（4）确认完成，将数据存入数据库中。

表5 常用脱密方法

|  |  |
| --- | --- |
| **脱密方法** | **详情** |
| 彻底删除法 | 对工艺技术和关键数据，彻底删除 |
| 模糊处理法 | 对特定的装备，可以省去装备具体名称，使用“某型”等或其他模糊名称；对某个具体数字，可以使用虚数表示 |
| 替换取代法 | 将保密的某个装备表示为“典型装备”或其他特定名字 |
| 数据示意法 | 对于具体某个装备性能的分数或参数，使用合理取值范围内其他数字表示 |

以上数据获取方式适用于适航领域软件文档，除此之外，本文还选择了几个在需求跟踪领域经常用到的公开数据集。在本文实验中能够使用这些数据集的主要原因包括以下三点：（1）适航领域软件文档主要用于工业部门，保密性较强，能够获取到的适航领域软件数据集数量较少，不足以支撑模型验证过程；（2）在适航领域软件需求跟踪任务中，主要关注软件文档中的系统需求、高级需求、低级需求和源代码，而在公开数据集上，软件文档也被分为几种类型，包括但不限于上述四种，因此公开数据集满足实验模型的输入要求；（3）在需求跟踪领域，已有研究者使用本文选择的公开数据集进行实验来评估提出方法的有效性，因此使用这些数据集更有利于本文设置对比实验，并且得到的对比结果更有说服力。

公开数据集可以在CoEST网站上获取到，包括：CM1-NASA、GANTT、eTOUR、iTrust和EasyClinic，其中CM1-NASA是美国国家航空航天局的基础项目CM1的一个子集；GANTT是一个使用甘特图来管理项目流程的软件；eTOUR是由Salerno大学使用java语言开发研制的一个专用于旅游的电子导航软件；iTrust是一个用于记录医药信息的java web系统；EasyClinic是用在医疗管理的java软件项目，同样由Salerno大学开发。每个数据集的具体情况如表6所示。CoEST网站不仅提供了测试数据，而且对每一组测试数据都给出了专业的跟踪链接。使用CoEST数据进行实验的优势包括：1）给验证模型的工作带来了极大的便利；2）保证了结果的准确性；3）减少了在学习排序阶段对数据的标注过程。



图15 数据获取过程示意图

表6 实验数据集

|  |  |  |
| --- | --- | --- |
| **数据集名称** | **包含的内容** | **词条（Token）数量** |
| CM1-NASA | 22个高级需求、53个低级需求、45个跟踪链接 | 5767 |
| GANTT | 17个高级需求、69个低级需求、68个跟踪链接 | 2080 |
| eTOUR | 58个用例、116个代码类、308个跟踪链接 | 97452 |
| iTrust | 131个用例、367个代码类、534个跟踪链接 | 123501 |
| EasyClinic | 30个用例、20个UML交互图、63个测试用例、47个类描述信息、1257个跟踪链接 | 21882 |

在确定软件数据集之后，还需要生成训练词向量的语料库。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。当然，在相同的领域，语料库越大效果越好。为了保证领域相关性，本文首先下载了GB/T11457-2006《信息技术软件工程术语》，然后去掉软件术语表中单个词的短语，接下来对每个软件术语添加上“software”，最后使用处理过的软件术语表去过滤维基百科数据集，最终得到一个包含有43,443,648个单词的264M的软件相关语料库。具体步骤如图16所示。



图16 获取训练词向量语料库流程

## 文档预处理

实验中文档根据类型可以分为两类：（1）文本文件，包括高级需求文档、低级需求文档、用例文档、UML交互图文档、测试用例文档、类描述文档等；（2）源代码文件。具体的文档预处理流程如图17所示。

第一步，设置领域词典。本文首先通过光学字符识别（Optical Character Recognition，OCR）将pdf版本的《汉英航空发动机工程技术词典》中的汉语和对应的英文解释提取出来，然后把汉语部分作为领域专有名词，并将汉语和英文部分存入数据库表中。

第二步，对中文软件文档进行分词和翻译。步骤如下：（1）使用结巴分词工具，并在分词时加入自定义词典，即第一步中的领域词典对文本进行分词；（2）去掉中文分词结果中的中文停用词；（3）分词得到的结果将优先使用领域词典对单词进行翻译，当领域词典中该词汇不存在时，使用有道云翻译接口进行翻译；（4）将最终的分词和翻译结果存入数据库中。

第三步，对分词并翻译之后的文本和代码文件进行以下处理：（1）以空格为分隔符对文本和代码内容进行分割；（2）去掉标点符号、数字和英文停用词。另外，在处理翻译后的文本时，使用的是nltk[46]，并在nltk提供的英文停用词列表中加入了一些数据集中无用的词；在处理代码文件时，我们将java和C语言中的关键字加入了停用词列表。

第四步，根据驼峰命名法的规则对复合名词进行分解。需要注意的是，这里只对不在软件术语表中出现的复合名词进行分解，即只分解由开发者命名的复合名词进行。举例说明，单词“DefaultMutableTreeNode”将被分解成四部分：“Default”、“Mutable”、“Tree”和“Node”。

第五步，将所有单词都转换成小写形式。需要注意的是，由于本文中使用word embedding，在训练使用的语料库中已经包含了单词的各种词形，因此提出的方法中并不需要对单词进行词干提取和词形还原。

最后，将以上的结果保存到数据库中。



图17 文档预处理流程

## 实验设置

正如第三章提到的，Tr-WELR模型中计算部分包括使用WQI方法计算文本语义相似度阶段和学习排序阶段。本文设置了三组实验，如表7所示，第一组实验是使用LSI[9]方法和公式(3.7)所示的文献[16]中计算文本相似度的方法W2V，分别与WQI方法做对比，用来检验提出的改进的文本语义相似度算法对结果的提升效果；第二组实验是使用改进的学习排序模型分别与未使用学习排序模型前的结果对比、未改进的IR SVM算法对比，用来验证学习排序算法在特定领域软件需求跟踪任务上的有效性和改进后算法性能的提升效果；第三部分实验是对比本文提出的Tr-WELR模型和当前国际领先水平的方法ENRL[17]，用来验证和评估本文提出的Tr-WELR模型的有效性和性能。下面将对以上提到的方法进行介绍。

表7 对比方法

|  |  |
| --- | --- |
| **实验** | **对比方法** |
| 实验一 | LSI、W2V |
| 实验二 | 未使用学习排序算法前的结果、IR SVM |
| 实验三 | ENRL |

在本文中，LSI方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，该方法已经被证明是一个在跟踪恢复领域成熟的方法[12]。该LSI方法步骤分为六步[9]，分别是：（I）定义一个潜在的跟踪模型；（II）在数据集上使用跟踪模型自动识别出文档概念；（III）预处理文档；（IV）重建跟踪链接；（V）选择相关的跟踪链接；（VI）可视化跟踪链接。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。

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。该对比方法的设置是为了从实验的角度说明改进的文本相似度算法的有效性和合理性。

改进的IR SVM算法首先按照Ranking SVM的方法把所有查询结果映射到文档对的空间中，将每一个文档对看作一个实例，使用SVM的方法把这些实例分成两类：+1和-1，其中+1表示文档对中前一个文档在结果列表中的顺序比后一个文档靠前，-1相反。在应用SVM对实例进行分类的过程中，使用了公式（3.16）所示的最优化公式，并且运用SMO算法来解决该最优化问题。

ENRL（Estimation of the Number of Remaining Links）方法是论文[17]中为了预测需求跟踪链接数量提出的模型。该模型结合了NLP方法和机器学习算法，并且为了探寻预测模型的精度，尝试了12种NLP方法和4个机器学习算法的组合，最终给出了每个数据集在所有组合上的结果。该论文中使用了三组数据，其中两组是本文同样使用到的eTOUR和EasyClinic，因此本文使用论文[17]中这两组数据最好的结果和Tr-WELR模型的结果对比。该实验的设置是为了验证提出的Tr-WELR模型的在软件需求跟踪任务上具有更好的效果。

本文使用主题模型工具gensim[47]在生成的语料库上训练词向量，同时选择word2vec训练模型中的CBOW模型，设置窗口大小为5，词向量的维度为200。

本论文中所有的实验都在相同的实验环境下进行，实验所用的机器为八核英特尔i7处理器，运行内存为8G。

## 模型评估指标

为了评估模型的性能，本文使用精确率、召回率、F测度、MAP（Mean Average Precision）和MRR（Mean Reciprocal Rank）为评估指标。在描述评估指标之前，记：

·为检索列表中与查询项有关联的文档数，即可以与查询的需求建立跟踪关系的文档的数量。

·为检索列表中与查询项无关联的文档数，即与查询的需求无法建立跟踪关系，却错误地与之建立跟踪关系的文档数。

·为数据集中与查询项没有关联的文档，同时也没有在检索列表中出现的文档数。

·为数据集中与查询项有关联的文档，但是不在检索列表中的文档数。

下面给出在软件需求跟踪领域中，常用的模型性能评估指标定义。

**精确率（Precision）**检索结果列表中与查询项有关联的文档占检索列表文档总数的比。

(4.1)

**召回率（Recall）**检索结果列表中与查询项有关联的文档数占数据集中所有与查询项有关联的文档数的比。

(4.2)

**F测度（F-measure）**好的模型应该同时有好的精确率和召回率，但是精确率和召回率有时候是矛盾的，即当精确率较高时，召回率低；当召回率较高时，精确率低。F测度用于计算精确率（Precesion）与召回率（Recall）的调和平均值，当F测度值较高时精确率和召回率的结果都相对较高。

(4.3)

**平均准确率（Average Precision）**平均准确率是对单个查询而言，即针对单个查询的结果集的准确率的平均值。

(4.4)

**MAP**该值为数据集合的平均准确率，是对所有独立查询的平均准确率（Average Precision）取平均值。举例说明：软件数据集中共有两个需求，req1和req2，两个需求对应的结果列表分别有4个和5个文档。与需求req1相关的文档在结果列表中的排名为1、2、4、7，与需求req2相关的文档在结果列表中的排名为1、3、5，则对于需求req1的平均准确率为（1/1 + 2/2 + 3/4 + 4/7）/ 4 = 0.83，对于需求req2的平均准确率为（1/1 + 2/3 + 3/5 + 0 + 0）/ 5 = 0.45，那么该软件数据集的MAP = （0.83 + 0.45）/ 2 = 0.64。

(4.5)

**MRR**与MAP所关心所有正确的文档位置不同，MRR仅关心第一个正确的文档的位置。对每个查询对应的候选文档列表，找到每一个集合中第一个正确的文档的位置，然后取倒数，再对所有集合中的倒数取平均值，如公式(4.6)所示。

(4.6)

其中为集合中第一个正确的文档的位置。举例说明：对于软件数据集共有两个需求，分别是需求req1和需求req2，需求req1建立跟踪关系的候选文档集合列表为doc1、doc2、doc3，其中doc2是与需求req1有跟踪关系的文档，需求req2对应的候选文档集合列表为doc4、doc5、doc6、doc7，其中doc6是与需求req2有跟踪关系的文档，则MRR = （1/2 + 1/3）/ 2 = 0.53。

**相对误差（Relative Error，RE）**相对误差是常用的用来评价预测模型准确率的指标之一[49]，在本文中该值可以表示为公式（4.7）所示。

(4.7)

**平均相对误差（Mean Relative Error，MRE）**平均相对误差同样用来评价预测模型的准确率，从全局的角度考虑方法的相对误差，计算方式如公式（4.8）所示。

(4.8)

## 实验结果与分析

这一节将详细描述三组实验的实验结果，根据结果对提出的适航领域软件需求跟踪算法模型的有效性和性能进行分析，并对一些性能提升点和异常情况进行详细的解释说明。

### 第一组实验：使用WQI算法

本组实验分别使用LSI方法、W2V方法和WQI算法对七组数据进行跟踪链接的恢复，并使用精确率、召回率和F测度进行模型评估，实验结果如表8所示。表8中PRE和REC分别表示精确率和召回率。HL、LL、UC、CC、ID、TC分别表示高级需求文档、低级需求文档、用例文档、类文件、UML交互图文档和测试用例文档。

在结果中可以看到，后两种使用词向量的方法W2V和WQI平均来说比LSI的效果更好。其中，WQI算法和LSI方法相比，WQI算法比LSI在精确率上相对提升了33.3%，在召回率上相对提升了24.5%；WQI算法和W2V方法相比，WQI算法比W2V方法在精确率上相对提升了6.6%，在召回率上相对提升了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 |

图18展示了三种方法在七组数据上F测度的平均值，显示了WQI算法比LSI方法和W2V方法在大多数数据集上效果都更好。

从表8中可以看出，WQI算法在七组不同类型的跟踪链接上，相比其他方法都有普遍的性能提升。跟踪链接能够根据软件文档的类型分为两类：一类是文本对文本（text to text，t2t），另一类文本对源代码（text to code，t2c）。表8中HL→LL、UC→ID和UC→TC都属于t2t，UC→CC属于t2c。同样，在图19中也能看出，WQI算法对两类跟踪链接的F测度都有提升。对于以上性能的提升，可以做如下解释：对于t2t类型的跟踪链接的性能提升，表示本文提出的改进的文本语义相似度算法WQI在软件需求跟踪任务是可行的，且效果不错；对于t2c类型的跟踪链接的性能提升，表示在实验之前对代码文件的预处理操作是合理的，且通过词向量表示时能够相对准确的表示其在代码中的实际含义。

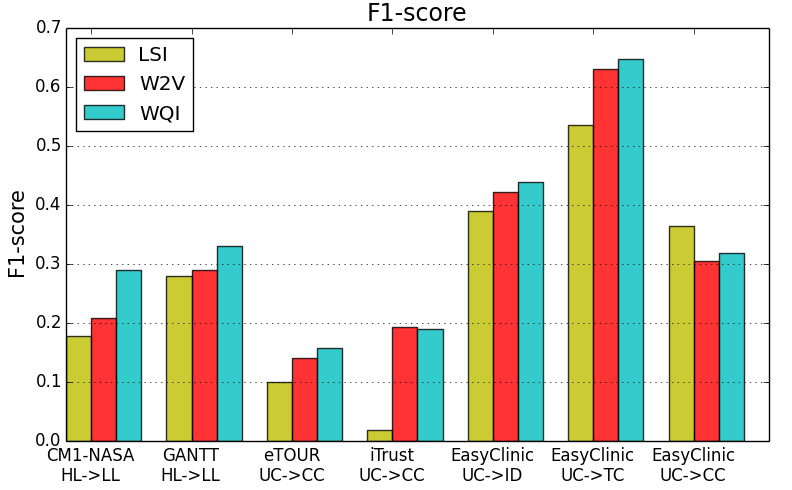


图18 LSI、W2V和WQI三种方法在七组数据上的F测度

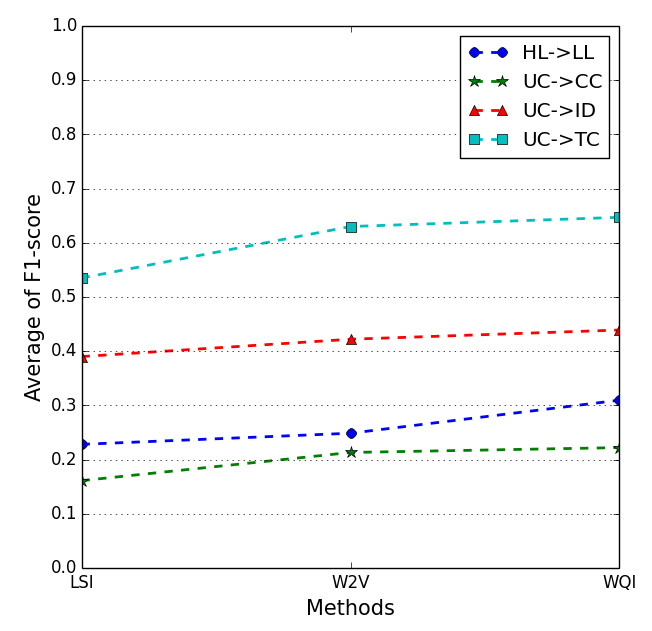


图19 LSI、W2V和WQI三种方法在不同类型的跟踪链接上的平均F测度

相比W2V方法，WQI算法引入了TFIDF加权策略和查询扩展技术，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。查询扩展技术使用和查询项相关或相似的单词或短语对查询项进行扩展，这也就在语义空间上丰富了查询项的含义[48]。从另一个角度来说，因为每一个词向量包含词的频率特征和上下文的信息，因此一组相似的词必定比一个特定的词包含的信息多。另外，权重策略TFIDF不仅仅能够减少句子中的一些噪声信息，也通过抽取句子中的关键词减少了待扩展的词，既提升了精度又提高了算法执行效率。

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，即生成的排序列表的质量更高。

### 第二组实验：使用改进的学习排序算法

学习排序算法将利用查询和排序列表的一些特征去学习、预测新的需求跟踪链接。本文使用学习排序算法去提升需求跟踪任务的精确度。为了验证学习排序算法是否可以让软件需求跟踪任务取得更好的效果，本节做实验去对比使用学习排序算法前后对结果的影响；同时为了验证改进的学习排序算法是否能够提升软件需求跟踪任务的性能，本节还将与未改进的IR SVM做对比试验。两个实验均使用MRR和MAP作为模型评估指标。学习排序算法使用之前介绍的五个特征去生成排序模型，同时做了十折的交叉验证，并且最终取这些验证的平均值作为最终结果。

图20展示了执行学习排序算法前后的MRR和MAP指标的值。和执行学习排序算法之前的结果相比，执行算法之后MRR和MAP的平均值都有所提升，具体来说，MRR的平均值提升了7.48%，MAP的平均值提升了22.05%。MAP指标的显著提升，表明IR SVM算法能够提升候选结果列表的整体排序水平和整体结果的精度；MRR指标的提升，说明在候选结果列表中首次出现正确文档的位置更靠前。从另一个角度来说，使用学习排序算法后，模型能够提供更准确的跟踪链接。

图21展示了改进的IR SVM学习排序模型和IR SVM模型的对比结果，同样使用MRR和MAP两个指标做对比。从图中可以看出改进之后的IR SVM在MRR值和MAP值上都有提升，具体来说，MRR平均相对提升了5.38%，MAP平均相对提升了9.80%。改进之后的IR SVM算法在实验所用的所有的数据集上的效果都有提升，从一定程度上说明改进的权重相比之前的权重机制有更好的效果，也说明了提出的观点——查询项与查询结果的语义相似度既包含了相对顺序关系也包含了决定相对顺序的可量化的测度——是比单独的使用相对顺序关系更有效的。

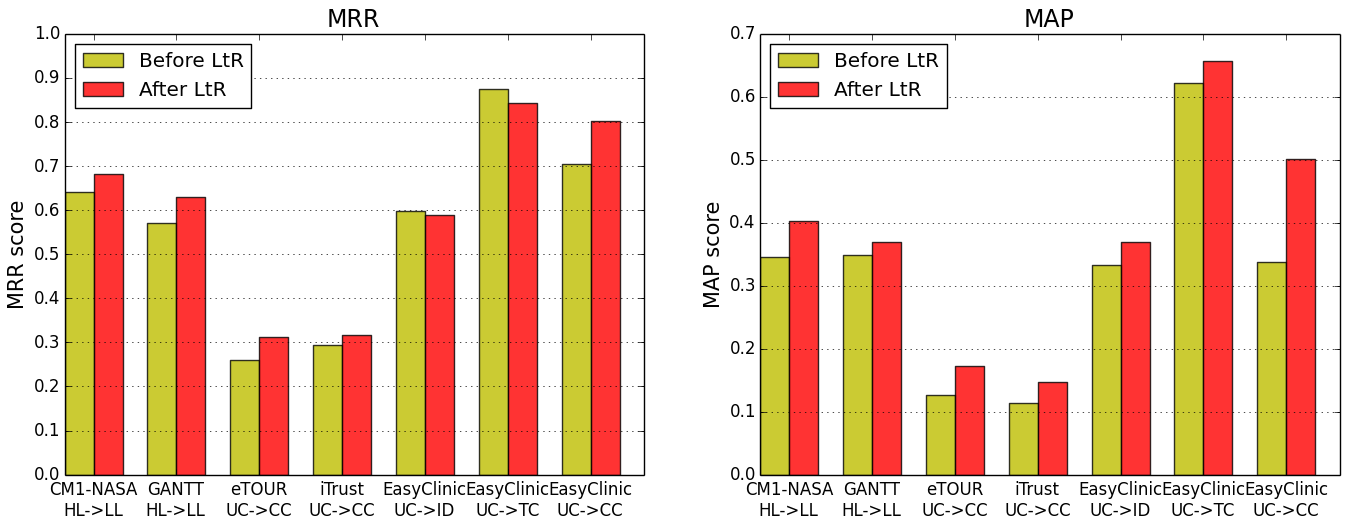


图20 执行改进的学习排序算法前后的MRR和MAP比较

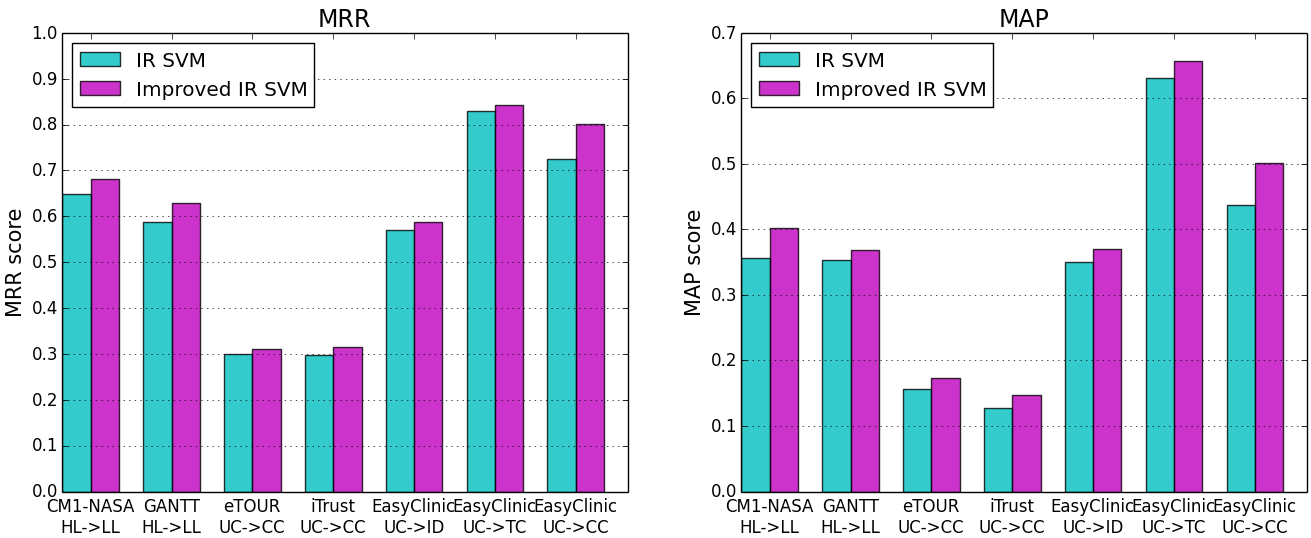


图21 IR SVM与改进后的IR SVM对比

### 第三组实验：Tr-WELR模型对比ENRL方法

本组实验采用和ENRL中相同的数据集eTOUR和EasyClinic，并且选择MRE作为评估指标。ENRL中尝试了12种不同的NLP方法和4种机器学习算法的所有组合，其中针对每个数据集的最优组合如表9所示。表中stop-stem表示去除停用词和对单词抽取词干，TF表示词频统计，LSA100表示使用潜在语义分析方法并设置概念数为100，VSM表示使用向量空间模型，cosine表示使用cosine函数计算相似度值，IBk表示使用k近邻方法分类实例。

表9 ENRL实验在eTOUR和EasyClinic上的最优组合

|  |  |  |
| --- | --- | --- |
| **数据集** | **NLP方法** | **机器学习分类算法** |
| eTOUR | stop-stem, TF, LSA100, cosine | IBk |
| EasyClinic | stop-stem, TF, VSM, cosine | IBk |

使用Tr-WELR模型对上述两个数据集计算，使用MRE作为评估指标，得到的结果与ENRL方法的对比如表10所示。从对比结果可以看出，在两个数据集上Tr-WELR模型计算得到的平均相对误差更小。

表10 Tr-WELR模型在eTOUR和EasyClinic上MRE值

|  |  |  |
| --- | --- | --- |
| **方法** | **数据集** | **MRE值** |
| ENRL | eTOUR | 0.0300 |
| EasyClinic | 0.0100 |
| Tr-WELR | eTOUR | **0.0208** |
| EasyClinic | **0.0059** |

综合前两组实验的分析，可以说明如下三点：

（1）Tr-WELR模型使用特殊训练的word embedding和改进的文本语义相似度算法，得到了比使用LSI和VSM等基于统计的信息检索方法更精确的文本相关度，同时在需求跟踪任务中，得到了准确率更高的需求跟踪链接。较好的结果说明在任务中使用的对文本的预处理和对参数的设置能够对实验的结果产生积极的影响，并且提出的文本语义相似度算法比其他的文本相似度算法效果更好，在需求跟踪任务中也是可行的。

（2）Tr-WELR模型使用基于文本语义相似度的权重表示方法改进了IR SVM方法中对不同等级（位置）的文档的惩罚因子，对比改进前后，该方法取得了较好的结果，说明改进的部分是有效的。

（3）对比国际领先水平的方法ENRL中对eTOUR和EasyClinic两个数据集的最优结果，Tr-WELR能够获得更小的平均相对误差。因此说明Tr-WELR在软件需求跟踪任务上具有更好的性能。

## 运行时间效率

表11显示了Tr-WELR模型中在不同数据集上计算部分的运行时间，其中计算部分包括文本语义相似度计算阶段和执行学习排序算法阶段。表中Mean R.T.(sec)表示在特定数据集上恢复一组跟踪链接需要的平均运行时间，Retrieval Nums指的是在每个数据集上对任意一个查询项需要检索的文档数，Mean Documents’ Length指的是每个数据集的平均文档长度。根据结果我们可以推断出平均运行时间和检索数目、平均文档长度都是正相关的，即需要检索的文档数越多，平均文档长度越长，需要的计算时间越长。

虽然本文实验中在检索数量最大的iTrust数据集上平均需要79.25秒才能建立全部的用例到代码类的跟踪关系，但是相比于人工建立需求跟踪关系，这个时间是完全可以接受的。

表11 Tr-WELR模型计算部分的运行时间和相关信息

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **数据集** | | **Mean R.T.(sec)** | **Retrieval Nums** | **Mean Documents’ Length** |
| CM1-NASA | HL→LL | 2.44 | 53 | 339 |
| GANTT | HL→LL | 2.05 | 69 | 107 |
| eTOUR | UC→CC | 49.42 | 116 | 1766 |
| iTrust | UC→CC | 79.25 | 367 | 2867 |
| EasyClinic | UC->ID | 18.09 | 20 | 604 |
| UC->TC | 27.00 | 63 | 604 |
| UC->CC | 29.78 | 30 | 604 |

## 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，主要原因有两方面：1）在软件文档上使用提出的基于word embedding的文本语义相似度算法准确率有所提高；2）使用学习排序算法使得跟踪结果的精度提高。