

# 安全关键软件术语推荐和需求分类方法

杨志斌<sup>1,2</sup> 杨永强<sup>1,2</sup> 袁胜浩<sup>1,2</sup> 周勇<sup>1,2</sup> 薛垒<sup>3</sup> 程高辉<sup>4</sup>

1 南京航空航天大学计算机科学与技术学院 南京 211106

2 高安全系统的软件开发与验证技术工信部重点实验室 南京 211106

3 上海航天电子技术研究所 上海 201109

4 北京控制与电子技术研究所 北京 100038

**摘要** 安全关键软件需求中的相关知识大多需要手工提取,既费时又费力。近年来,人工智能技术逐渐被应用于安全关键软件设计与开发过程中,以减少工程师的手工劳动,缩短软件开发生命周期。文中提出了一种安全关键软件术语推荐和需求分类方法,为安全关键软件需求规约提供了基础。首先,基于词性规则和依存句法规则对候选术语进行提取,通过术语相似度计算和聚类方法对候选术语进行聚类,将聚类结果推荐给工程师;其次,基于特征提取方法和分类方法将安全关键软件需求自动分为功能、安全性、可靠性等需求;最后,在AADL(Architecture Analysis and Design Language)开源建模环境OSATE中实现了原型工具TRRC4SCSTool,并基于工业界案例需求、安全分析与认证标准等构建实验数据集进行了实验验证,证明了所提方法的有效性。

**关键词:** 安全关键软件;需求工程;术语推荐;术语聚类;需求分类

**中图分类号** TP311

## Terminology Recommendation and Requirement Classification Method for Safety-critical Software

YANG Zhi-bin<sup>1,2</sup>, YANG Yong-qiang<sup>1,2</sup>, YUAN Sheng-hao<sup>1,2</sup>, ZHOU Yong<sup>1,2</sup>, XUE Lei<sup>3</sup> and CHENG Gao-hui<sup>4</sup>

1 School of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China

2 Key Laboratory of Safety-critical Software, Ministry of Industry and Information Technology, Nanjing 211106, China

3 Shanghai Aerospace Electronic Technology Institute, Shanghai 201109, China

4 Beijing Institute of Control and Electronic Technology, Beijing 100038, China

**Abstract** Most of the knowledge in the requirements of safety-critical software needs to be manually extracted, which is time-consuming and laborious. Recently, artificial intelligence technology has been gradually used in the design and development of safety-critical software, to reduce the work of engineers and shorten the life cycle of software development. This paper proposes a terminology recommendation and requirement classification method for safety-critical software. Firstly, the terminology recommendation method extracts candidate terms based on part-of-speech rules and dependency rules and clusters candidate terms through term similarity calculation. The clustering results are recommended to engineers. Secondly, the requirement classification method automatically classifies safety-critical software requirements as functional, safety, reliability, etc. based on feature extraction. Finally, the prototype tool TRRC4SCSTool is implemented in the AADL open-source modeling environment OSATE, and the experimental analysis is carried out through the dataset collected from the industrial requirements and safety certification standards, and the results show the effectiveness of the method.

**Keywords** Safety-critical software, Requirement engineering, Term recommendation, Term clustering, Requirement classification

## 1 引言

安全关键软件(Safety-Critical Software, SCS)<sup>[1]</sup>指应用

于航空、航天等领域的安全关键系统中,且其运行情况可能引起系统处于危险状态,从而导致财产损失、环境破坏或者人员伤亡的一类软件,它对功能正确性、实时性、安全性等性质有

到稿日期:2021-01-13 返修日期:2021-03-27

基金项目:国家自然科学基金(62072233);航空科学基金(201919052002);国防基础科研项目(JCKY2020205C006)

This work was supported by the National Natural Science Foundation of China (62072233), Aeronautical Science Foundation of China (201919052002) and Defense Industrial Technology Development Program (JCKY2020205C006).

通信作者:杨志斌(yangzhibin168@163.com)

极高的要求。这类软件往往都需要遵循严格的设计与开发标准,例如民用机载软件开发国际标准 DO-178C<sup>[2]</sup>。

已有研究表明,在很多情况下,安全关键软件引起严重事故的问题链的最上端原因往往是软件需求尤其是安全性需求的问题<sup>[1-3]</sup>,例如需求规约不完整或安全需求缺失。需求工程(Requirement Engineering, RE)已经成为安全关键软件领域的一个重要研究内容,主要涉及需求捕获与分析、规约、验证、维护等阶段。相比通用软件,安全关键软件需求工程的主要不同之处在于安全需求的捕获、分析及规约。在获得正常功能需求之后,通常需要按照相关标准进行安全分析,例如民用机载系统安全性评估标准 ARP4761,常常基于故障树(Fault Tree Analysis, FTA)、失效模式影响分析(Failure Mode and Effects Analysis, FMEA)、依赖图(Dependence Diagram, DD)、马尔可夫分析(Markov Analysis, MA)等方法进行安全分析,从而获得安全需求。

目前,安全关键软件需求规约主要通过自然语言文本、限定自然语言文本(如限定自然语言需求模板)<sup>[4-5]</sup>或 SysML/UML 用例图等方式来实现。安全关键软件需求规约文档主要包括领域词库、数据字典、需求规定等内容。领域词库和数据字典也称为术语定义(glossary),是为了确保需求表达的一致性,实现对需求规约文档中的数据、对象名词的统一集中管理和显式规范描述。在需求规定中,一般将需求分为功能需求、安全需求、性能需求、可靠性需求、可维护性需求、保密性要求、接口需求等。随着安全关键软件系统越来越复杂,需求术语提取和自然语言需求分类成为了一个重要挑战,主要原因是:需求相关方(用户、需求工程师、系统工程师、安全工程师、测试工程师等)对同一类需求可能使用不同的术语和需求语句表达格式,这种不一致性导致需求术语手工提取和分类非常耗时且容易出错<sup>[6]</sup>。

近年来,引入人工智能技术来辅助需求术语提取和分类成为了一个热点研究。自动术语提取研究可分为 3 种方式:语言学、统计和混合<sup>[7]</sup>。所谓语言学方法,即根据语言特性来制定提取术语的规则模板,其中规则模板主要根据词法、语法等语言学规则和领域特征构建而成;统计学方法则是根据统计指标(如频率和长度)选择术语;针对语言学和统计两种方法的不同特点,两种方法混合使用成为了重要选择。Arora 等在基于语言学的术语提取研究的基础上,进一步提出了术语聚类方法<sup>[8]</sup>。在自然语言需求分类方面,采用的机器学习方法包括监督学习、半监督学习和无监督学习这 3 类。Kurtanovic 等<sup>[9]</sup>提出基于支持向量机(Support Vector Machine, SVM)的需求分类方法,将需求分类为功能需求和非功能需求;Sunner 等<sup>[10]</sup>使用基于遗传算法的神经网络,实现功能需求和非功能需求的分类;Canedo 等给出了用于需求分类的不同特征提取方法和机器学习方法的比较<sup>[6]</sup>。另外,针对需求规约文档中经常存在的除需求以外的辅助信息,Abualhaija 等提出了基于机器学习的需求和辅助信息的分离(demarcation)方法<sup>[11]</sup>。这些已有研究主要针对英文自然语言需求,中文词语间无空格分隔、歧义现象普遍、实词运用灵活等特点,形成了其特有的复杂性和难度。针对中文需求,Jia 等<sup>[12]</sup>提出了一种自动化的非功能需求识别和分类方法,其分类方法

主要基于语义距离和相似度计算,将包含非功能需求语句划分为性能、可靠性、可用性、安全性、可维护性等 5 类。

然而,已有研究主要针对通用软件领域,目前较少考虑安全关键软件系统,因此本文提出了一种安全关键软件术语推荐和需求分类方法。相比已有研究,本文研究的不同之处主要有:针对中文自然语言文本进行研究,尽量不改变国内工程师撰写需求的习惯;考虑安全关键领域特征,将工业界实际案例需求、安全分析与认证标准、安全分析报告等作为实验数据集;将术语推荐和需求分类紧密结合起来,即将所推荐的术语作为需求分类中特征提取的重要基础。本文的主要贡献包括:

(1)提出了一种面向安全关键软件的术语推荐方法,主要包括候选术语提取、术语相似度计算和候选术语聚类。首先,基于词性标注和依存句法分析等自然语言处理技术制定完整的术语推荐规则集(包括词性规则及依存句法规则)和过滤指标来抽取候选术语;然后,计算候选术语对之间的相似度并生成相似矩阵;最后,使用相似矩阵和聚类算法来聚类候选术语。

(2)提出了一种面向安全关键软件的需求分类方法,主要包括特征提取和需求分类。特征提取阶段将需求文本转化为向量形式,需求分类阶段使用机器学习分类算法对安全关键软件需求进行分类。

(3)设计并实现了 TRRC4SCSTool 原型工具,并将工业界实际案例需求、安全分析与认证标准、安全分析报告等作为实验数据集,设计实验来验证所提方法的可用性和有效性。

## 2 术语推荐和需求分类方法

### 2.1 总体框架与示例

总体框架如图 1 所示,包括术语推荐和需求分类两部分,两者共同为需求规约提供基础。在术语推荐方面,将安全关键软件需求、安全分析与认证标准、安全分析报告等作为输入,经过候选术语提取、术语相似度计算和候选术语聚类 3 个步骤来获得候选术语表;在需求分类方面,针对输入数据集,经过文本预处理、提取文本特征、训练分类器和评估分类器 4 个步骤处理后得到分类器,并将安全关键软件需求输入分类器,进而得到可能的需求类别。要注意的是,由于候选术语是基于软件需求、安全分析与认证标准、安全分析报告等文本进行提取的,因此将候选术语引入特征提取过程中,以帮助提升需求分类方法的准确度。

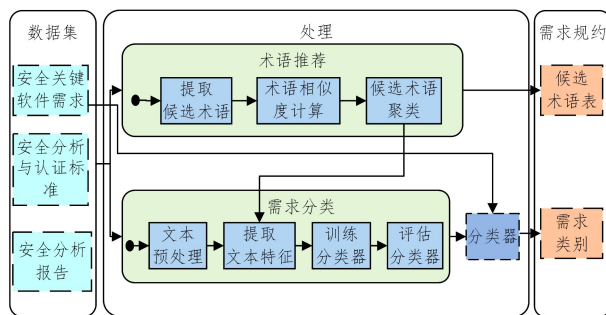


图 1 总体框架  
Fig. 1 Global view

我们通过一个运行示例来介绍本文方法的总体效果,如

图2所示。图2(a)给出了航天发射控制子系统的7个安全关键软件需求。图2(b)给出了术语推荐方法的结果。按照本文术语推荐方法的3个步骤进行描述:首先提取需求中的候选术语,例如R1中提取出“通用化发控系统”“发控单元”“数字数据总线”和“输入信号波形”4个候选术语;其次计算候选术语的相似度,例如使用Dice系数相似度计算策略,“发控电路”与“发控接口”的语法相似度为0.571429,而“发控电路”

与“地址回线”的语法相似度为0.0;最后根据候选术语的相似度生成的相似矩阵进行聚类。例如,聚类算法会将“发控电路”“发控接口”“发控单元”与“发控检查标志”4个候选术语聚为一类。图2(c)给出了需求分类方法的结果,即将需求文本分类为功能、性能等5类需求。例如,R6中存在时间约束条件,可以归为性能需求;R3描述系统的功能行为,可以归类为功能需求。

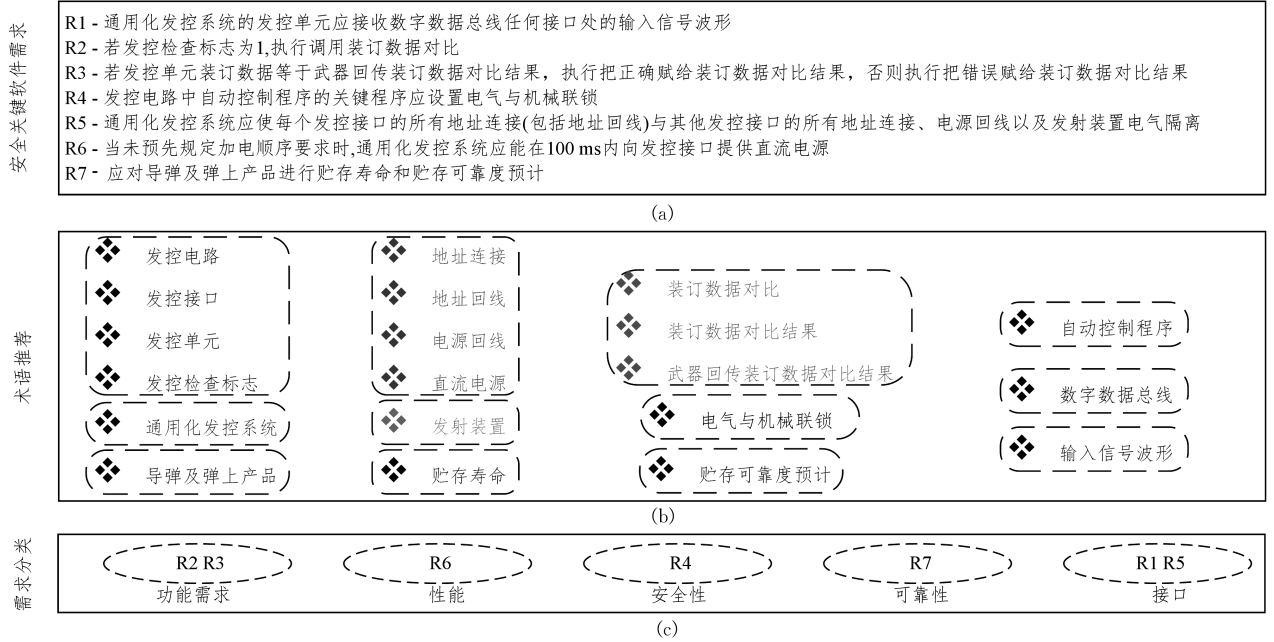


图2 运行示例

Fig. 2 Running example

2.2 安全关键软件术语推荐方法 TR4SCS

本节给出了面向安全关键软件的术语推荐方法(Terminology Recommendation for Safety-Critical Software, TR4SCS), 如图3所示。

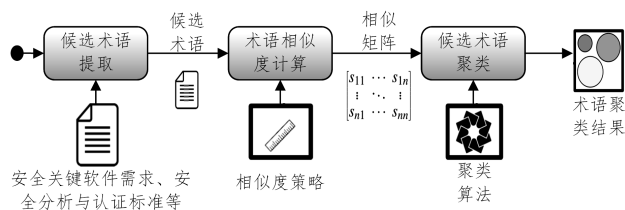


图3 安全关键软件术语推荐方法流程

Fig. 3 TR4SCS process

给定安全关键软件需求等文本,首先提取候选术语列表,然后根据术语相似度策略计算候选术语的相似矩阵,最后根据相似矩阵对候选术语进行聚类。

2.2.1 候选术语提取

候选术语提取方法包括文本预处理、词性规则提取、依存

句法规则提取、领域度过滤4个步骤,如图4所示。为了尽可能地降低人工检查的范围,此步骤重点关注候选术语提取的召回率。

(1) 文本预处理

对安全关键软件需求、安全分析报告等文本进行分句、分词、词性标注、依存句法分析等处理。该步骤基于已有的自然语言处理工具 HanLP<sup>[13]</sup> (Han Language Processing)来实现。图5给出了“飞机管理计算机通过远程接口单元,发送启动空气压机的指令给空气增压机控制器”需求的词性标注和依存句法分析结果。

(2) 词性规则提取

安全关键软件领域术语存在语言结构特征。从外部关联来看,领域术语大多是名词性短语,其经常作为领域文本中的主语、宾语、定语等成分;从其内部语法构成来看,其组成形式包括名词+名词(“安执机构”)、形容词+名词(“可燃材料”)、动词+名词(“发射装置”)、动词(名词)+单字名词(“气密门”)等。领域术语长度主要是2~6个词。

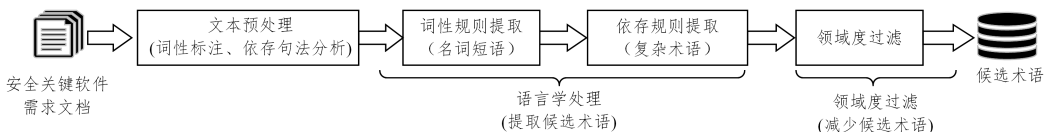


图4 候选术语提取

Fig. 4 Candidate term extraction

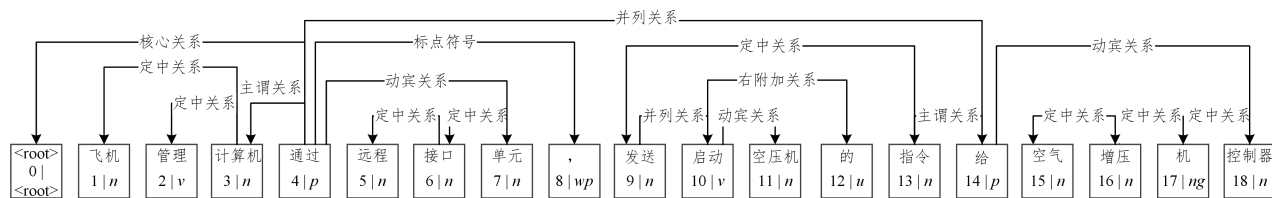


图5 词性标注和依存句法分析示例

Fig. 5 Example of POS tag and parsing

我们根据领域术语特征制定词性模式匹配规则,如表 1 所列。其中,  $n, vn, p, w, v, ng, b, gi, a, vi$  和  $nz$  分别对应的词性为名词、名动词、介词、标点符号、动词、名词性语素、区别词、计算机相关词、形容词、不及物动词和其他专有名词。例如“飞机管理计算机通过远程接口单元,发送启动空气压缩机的指令给空气增压机控制器”需求中,符合一词的词性匹配规则“ $n \& length > 3$ ”的候选术语有“计算机”“空压机”和“控制器”,符合二词的词性匹配规则“ $n + vn, vn + n, n + n$ ”的候选术语有“飞机管理”“管理计算机”“远程接口”“接口单元”和“空气增压”,符合三词的词性匹配规则“ $n + vn + n, n + n + n$ ”的候选术语有“飞机管理计算机”和“远程接口单元”,符合四词的词性匹配规则“ $n + n + ng + n$ ”的候选术语有“空气增压机控制器”。

表 1 词性模式匹配规则

Table 1 Part of speech pattern matching rules

Length	POS pattern matching rules
One	$n \& length > 3$
Two	$n + n, n + v, v + n, a + n, b + n, vn + n, n + vn$
Three	$n + n + n, v + n + n, n + v + n, v + v + n, v + n + v, b + n + n, m + n + v, n + nz + n, n + vn + n, n + vn + n, n + v + vn, n + n + vn, gi + n + n$
Four	$n + n + n + n, n + n + v + n, v + n + n + n, v + v + n + n, b + n + n + n, v + n + vn + n, n + v + a + n, n + n + ng + n$
Five	$v + v + n + n + n, n + n + n + vn + n$
Six	$n + v + a + n + vn + n, n + n + n + vi + vn + n, n + gi + a + n + vn + n, n + gi + n + n + vn + n$

(3) 依存句法规则提取

依存句法分析通过分析语句中词语之间的依存关系,揭示语句的句法结构。其中,依存关系可以使用有向弧表示,由支配词指向被支配词,并且依存句法分析认为语句中的支配者是核心动词。在一个完整的语句中,依存句法分析将语句的线性结构层次化,构造成为依存树。依存树的定义如下。

**定义 1 (依存树)** 依存树记为  $T = (V, A, R)$ 。其中,  $V$  为结点集合,表示句子中的词语;  $A$  为有向弧集合,表示词语间的依存关系,弧的出发端为依存关系的支配词,弧的指向端为依存关系的被支配词;  $R$  为依存树根结点,是语句的核心动词,  $T$  满足:

- 1)  $R$  结点的入度为 0;
- 2) 除  $R$  之外的结点的入度为 1;
- 3) 从  $R$  到任一结点有一条有向通路。

依存句法规则提取包括生成依存树、剪枝和生成依存子树 3 个步骤<sup>[14]</sup>。首先,对文本进行依存句法分析,得到依存树;然后,依存树中一些依存关系通常不会出现名词短语,而领域术语一般为名词短语,因此我们对依存关系进行限定,对

依存树进行剪枝。本文主要保留定中关系 (ATT)、并列关系 (COO)、右附加关系 (RAD) 和左附加关系 (LAD) 4 类依存关系;最后,根据定义 2,对剪枝后的依存树进行筛选,共生成“管理计算机”“飞机管理计算机”“远程接口”“接口单元”“远程接口单元”“空气增压”“增压机”“空气增压机”“机控制器”“增压机控制器”和“空气增压机控制器”等依存子树。根据生成的依存子树,本文选取  $|V| > 1$  且连续的词串作为候选术语。例如,“飞机管理”“远程单元”为非连续词串,而剩余的依存子树均为连续词串,因此生成 11 个候选术语,分别为“管理计算机”“飞机管理计算机”“远程接口”“接口单元”“远程接口单元”“空气增压”“增压机”“空气增压机”“机控制器”“增压机控制器”和“空气增压机控制器”。

**定义 2 (依存子树)** 给定一棵依存树  $T = (V, A, R)$ , 则依存子树  $T' = (V', A', R')$  满足如下条件:

- 1)  $V' \subseteq V, A' \subseteq A, R' \subseteq R$ ;
- 2)  $R'$  结点的入度为 0;
- 3) 除  $R'$  之外的结点的入度为 1;
- 4) 从  $R'$  到任一结点有一条有向通路;
- 5)  $R'$  为名词或动词等实词。

依存句法规则提取算法如算法 1 所示。首先,对每个需求进行依存分析并生成依存树;然后,生成仅包含定中关系 (ATT)、并列关系 (COO)、左附加关系 (LAD)、右附加关系 (RAD) 4 类依存关系的依存子树;最后,分析每个依存子树中的每个字符串,提取出连续且长度大于 1 的字符串作为候选术语。此算法的时间复杂度和空间复杂度均为  $O(n^3)$ 。

**算法 1** 基于依存规则的术语提取算法

```

输入:需求文本 Requirements
输出:候选术语 CandidateTerms
1. for each Requirement in Requirements
2. DT ← ComputeDependencyTree(Requirement);
3. DSTs ← ComputeDependencySubTrees(DT); /* 计算仅包含 ATT, COO, LAD, RAD4 类依存关系的依存子树 */
4. for each DST in DSTs
5. for String in DST
6. if String is continuous and |String| > 1
7. CandidataTerms.add(String); /* 将词串加入到候选术语 */
8. end if
9. end for
10. end for
11. end for
    
```

(4) 领域度过滤  
术语的领域特性表示该术语通常仅出现在一个或几个特

定领域的文本中,并且仅在该特定领域中使用,而通用词对于所有使用该语言的人都是通用的。因此,本文共设计两种领域度过滤策略来过滤候选术语。策略1通过比较需求语料库中候选术语的出现次数与非特定文本语料库(如THUC-News)中文文本分类数据集的子集中出现的次数来确定过滤候选术语。策略2对存在通用词的候选术语进行过滤,如“的”“应”“以便”等。针对“飞机管理计算机通过远程接口单元,发送启动空气压机的指令给空气增压机控制器”这条需求,共提取出14个候选术语,分别为“飞机管理”“管理计算机”“飞机管理计算机”“远程接口”“接口单元”“远程接口单元”“空气增压”“增压机”“空气增压机”“机控制器”“增压机控制

器”“空气增压机控制器”“空压机”和“控制器”。

### 2.2.2 术语相似度计算

为了计算候选术语的相似矩阵,我们考虑3种策略:

(1)基于语法相似度,即仅使用语法相似度策略计算每一对候选术语( $t, t'$ )的相似度 $S_{syn}(t, t')$ 。本文采用街区距离(block distance)<sup>[15]</sup>、余弦相似度(cosine)<sup>[15]</sup>、Dice系数(Dice's coefficient)<sup>[15]</sup>、编辑距离(Levenshtein)<sup>[16]</sup>、欧几里得距离(Euclidean)<sup>[15]</sup>、Jaccard相似性系数(Jaccard)<sup>[17]</sup>、Jaro距离(Jaro)<sup>[17]</sup>、Jaro-Winkler距离(Jaro-Winkler)<sup>[17]</sup>以及SimHash-Hamming Distance<sup>[18]</sup>这9种算法来计算术语对的语法相似度。语法相似度的算法描述如表2所列。

表2 语法相似度的算法描述

Table 2 Grammatical similarity algorithm description

名称	描述
Block Distance	又称曼哈顿距离、车厢距离、绝对值距离、L1距离和街区距离。通过将术语转化为向量并且计算两个向量在标准坐标系上的绝对轴距总和来评估术语对的相似度
Consigne	通过将术语转换为向量并且计算向量之间的夹角余弦值来评估术语对的相似度
Dice's coefficient	通过查找共同字符,然后将共同字符的数量除以术语中标记的总数,来计算术语对的相似度
Levenshtein	根据将一个术语转换为另一个术语所需的最少字符编辑(插入、删除和替换)次数来计算术语对的相似度
Euclidean	通过将术语转换为向量并计算它们之间的归一化差来计算术语对的相似度
Jaccard	通过计算两个集合交集的大小除以并集的大小来评估术语对的相似度
Jaro	使用术语的每个标记中的公共字符数来计算术语之间的相似度
Jaro-Winkler	Jaro相似度的扩展,它将Jaro得分与术语之间的公共前缀长度结合在一起
SimHash-Hamming Distance	先使用SimHash把不同长度的文本映射为等长文本,然后计算等长文本的汉明距离

(2)基于语义相似度,即仅使用语义相似度策略计算每一对候选术语( $t, t'$ )的相似度 $S_{sem}(t, t')$ 。我们采用基于知网(HowNet)和同义词词林(Cilin)这两种算法来计算候选术语对的语义相似度。

(3)基于语法和语义相似度,即给定一对候选术语( $t, t'$ )的语法相似度 $S_{syn}(t, t')$ 和语义相似度 $S_{sem}(t, t')$ ,我们选取 $\max(S_{syn}(t, t'), S_{sem}(t, t'))$ 作为候选术语对的相似度。

### 2.2.3 候选术语聚类

候选术语聚类过程的输入包括相似矩阵、聚类算法以及聚类个数 $K$ ,输出为 $K$ 类候选术语的数组。为了选择较好的

聚类算法,本文采用 $K$ 均值( $k$ -means)、期望最大化(expectation-maximization, EM)和层次聚类(hierarchical clustering)3种聚类算法来进行聚类。本文采用不同相似度计算策略和聚类算法的组合来进行对比实验,并选取效果较好的可选组合作为工具的默认算法。

## 2.3 安全关键软件需求分类方法 RC4SCS

本节给出面向安全关键软件的需求分类方法(Requirements Classification for Safety-Critical Software, RC4SCS),其流程如图6所示,包括收集数据集、文本预处理、选取文本特征、训练分类模型和评估分类模型5个主要步骤。

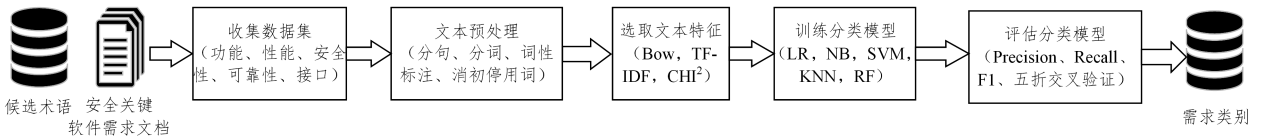


图6 安全关键软件需求分类方法流程

Fig. 6 RC4SCS process

### (1)收集数据集

首先,依据国家军用标准《军用软件需求分析》(GJB 1091-91)中的描述:需求分析文档分为功能、性能、接口、数据、环境、安全和保密、可修改性和假设与约束等8种需求类别;其次,依据《航天产品可靠性保证要求》(Q/QJA 297-2014)和《军用软件可靠性评估指南》(GJB/Z 161-2012)等对航天产品及军用软件的可靠性指标做出要求。本文挑选5种典型的需求类别来收集数据集,分别是功能、性能、

安全性、可靠性和接口。

### (2)文本预处理

对需求文本执行分词、词性标注和消除停用词等操作。首先,基于候选术语对需求文本进行分词和词性标注,该工作基于已有的自然语言处理工具HanLP来完成;其次,为消除停用词(停用词指一些非常常见的词),去掉它们一般不影响需求文本的语义。消除停用词有助于减少机器学习算法的文本特征数量。

(3)选取文本特征

将需求文本表达成向量的形式,常见的文本特征模型包括词袋模型(Bag-of-Words, Bow)和TF-IDF(Term Frequency-Inverse Document Frequency)模型。

Bow模型将安全关键软件需求数据集中不同词语组成的词典(包括推荐的候选术语)作为文本特征,将词语频率(即词语在需求文本中出现的次数)作为文本特征的特征值。为了解决Bow模型存在高频词汇会拥有较大的特征值的问题,TF-IDF模型通过统计词汇在所有需求文本中出现的频率来调整文本特征的特征值,以便对高频词汇的得分进行惩罚。TF-IDF的计算公式如式(1)所示:

$$TFIDF_{d,t} = TF_{d,t} \times IDF_t \quad (1)$$

其中,TF指某一个给定的词语在该文件中出现的频率;TF<sub>d,t</sub>指词语t在文档d中出现的频率;IDF指逆向文档频率;IDF<sub>t</sub>指某一特定词语t的逆向文档频率。表3列出了5类需求中前10个最关联的关键词。

表3 各类需求的前10个关键词

Table 3 Top 10 keywords of different types of requirement

功能	性能	安全性	可靠性	接口
回线	距离	安全	可靠性	接口
武器	功率	危险	失效率	连接器
直流电源	起飞	关键	寿命	线缆
执行	低温	安全性	应力	通信
锁定	燃油	身份认证	极限条件	互连
解锁	高度	中断	试验应力	双绞屏蔽线
通用化	速度	应急	维修	数据电路
报文	利用率	报警	强度	光缆
调用	重量	有效载荷	耐烧蚀	电缆
电控系统	负荷	泄漏	缺失率	数据接口

(4)训练分类模型

将文本特征的向量矩阵以及所属类别作为输入,使用机器学习分类算法训练分类模型。为了选择效果较好的分类模型,本文采用逻辑回归(Logistic Regression, LR)、朴素贝叶斯(Naive Bayes, NB)、SVM、K近邻(K-Nearest Neighbors, KNN)、随机森林(Random Forest, RF)5种典型的机器学习分类算法进行实验。

(5)评估分类模型

将测试文本的特征向量矩阵作为分类模型的输入,分类模型会预测出测试文本所属的类别。本文采用五折交叉验证的方式评估分类模型,评估指标采用传统的机器学习分类指标,即精确率(Precision)、召回率(Recall)和F1-measure,具体计算方式见第4.3节。

基于候选术语的需求分类算法如算法2所示,此算法的时间复杂度和空间复杂度均为O(n<sup>2</sup>)。

算法2 基于候选术语的需求分类算法

输入:需求文本 Requirements;候选术语 CandidateTerms

输出:需求类别 RequirementCategory

1. Tokenizers←Tokenizer(Requirements, CandidateTerms); /\* 基于候选术语特征,对训练集的需求文本进行分词 \*/
2. Vectorizers←Vectorizer(Tokenizers);
3. Classifiers←Classifier(Vectorizers);

4. Evaluators, RequirementCategory←Evaluator(Classifiers, Requirements, CandidateTerms). /\* 评估测试集的需求文本的性能 \*/

3 工具支持

3.1 工具框架

TRRC4SCSTool(Terminology Recommendation and Requirements Classification for Safety-Critical Software Tool)是基于Java程序语言开发并以扩展插件的形式集成到开源AADL(Architecture Analysis and Design Language)<sup>1)</sup>工具环境<sup>[19]</sup>(Open Source AADL Toolkit Environment, OSATE)中。TRRC4SCSTool原型工具的实现主要遵循MVC(Model-View-Controller)的设计思想,工具实现框架如图7所示。

(1)View Package,主要包括需求展示(RequirementView Package)、术语推荐展示(TermRecommendationView Package)和需求分类展示(RequirementCategoryView Package)。

(2)Model Package主要包括3种模型:安全关键软件需求模型(SCSReq Package)、术语推荐模型(TermRecommendation Package)和需求类别模型(RequirementCategory Package)。其中,SCSReq Package实现对安全关键软件需求的数据模型定义;TermRecommendation Package包括3个部分,CandidateTerm Package实现对候选术语的数据模型定义,SimilarityMatrix Package实现对候选术语相似矩阵的数据模型定义,TermClustering Package实现对术语聚类结果的数据模型定义;RequirementCategory Package则实现对需求类别的数据模型定义。

(3)Controller Package主要包括4种控制器:候选术语提取控制器(TermExtraction Package)、相似度计算控制器(SimilarityCalculation Package)、候选术语聚类控制器(Term-Clustering Package)及需求分类控制器(RequirementClassification Package)。Term-Extraction Package使用HanLP来完成术语抽取。SimilarityCalculation Package调用Similarity来实现相似度计算,并将Jaro-Winkler距离作为语法相似度默认策略,知网相似度作为语义相似度默认策略。Term-Clustering Package调用Weka机器学习包来实现术语聚类。使用层次聚类(hierarchical clustering)算法作为聚类算法的默认策略。RequirementClassification Package同样使用HanLP和Weka来实现需求分类。本步骤将词袋模型作为文本特征提取的默认策略,将支持向量机算法作为分类模型默认策略。工具的主要功能代码统计如表4所列。

表4 工具的代码统计

Table 4 Statistical code data of the prototype tool

模块	功能	功能解释	java(行)
前端	Interface	外部接口	300+
	Term Extraction	提取候选术语	600+
	Similarity Calculation	计算候选术语的相似度	300+
后端	Term Clustering	候选术语聚类	100+
	Requirement Classification	需求分类	500+

<sup>1)</sup> AADL是嵌入式软件体系结构建模语言标准

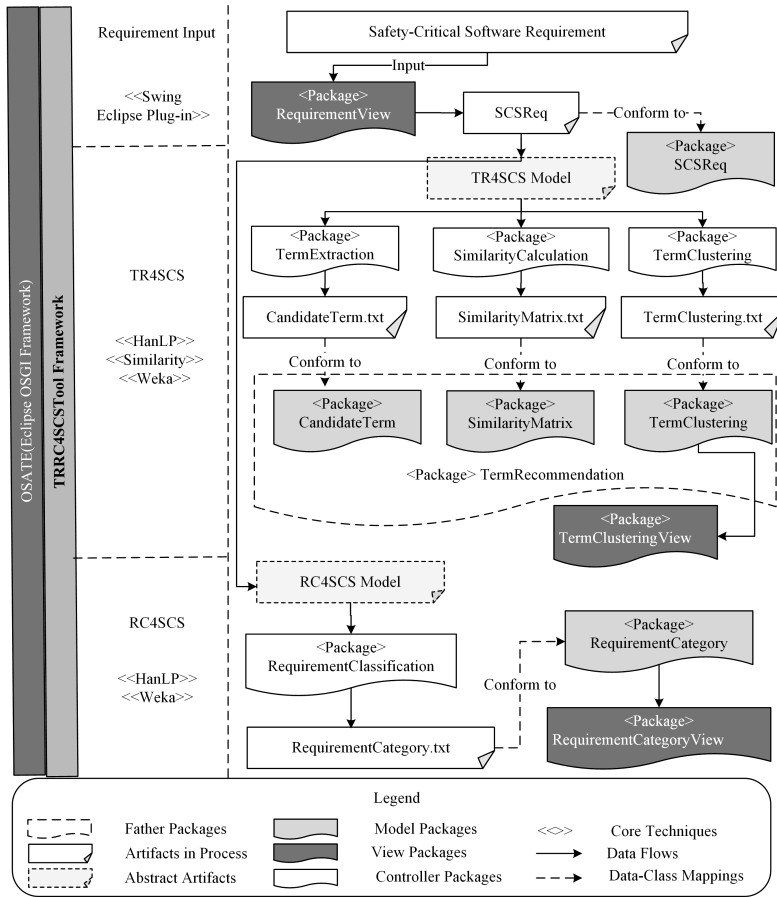


图 7 工具框架

Fig. 7 Tool framework

3.2 工具实现

本节主要给出术语推荐模块和需求分类模块的实现效果。

术语推荐模块的实现效果如图 8 所示,左侧面板展示术

语聚类,每个聚类都表示为注释类型。当选择一个聚类(注释类型)时,该聚类中的所有候选术语都会在右侧面板的文档中突出显示。界面的右下角包含加入数据字典和加入领域词库两个功能,可以将当前选中的聚类输出到对应的文件中。



图 8 术语推荐模块实现

Fig. 8 Terminology recommendation module implementation

需求分类模块的实现效果如图 9 所示。与术语推荐模块类似,每个需求类别表示一种注释类型。选择一个需求类别

(注释类型)时,其中所有需求语句都会在文档中高亮显示。用户可点击加入需求模板,将选中的需求语句输出到对应文件。

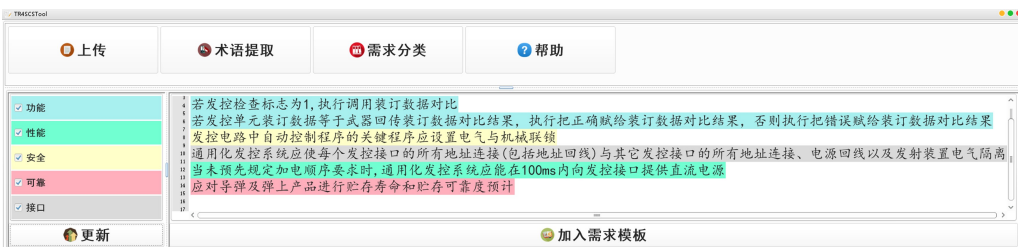


图 9 需求分类模块实现

Fig. 9 Requirements classification module implementation

## 4 实验验证

首先,阐述研究问题、实验数据集以及评估指标;其次,分别介绍研究问题的实验设计、结果和对比;最后,对本文方法和实验进行有效性威胁分析。

### 4.1 研究问题

本文的实验验证旨在回答以下研究问题。

(1)RQ1。面向安全关键软件的术语推荐方法的实用性如何? RQ1 旨在探究本文方法的实用性,以及在实际应用中与人工术语提取时间的对比。

(2)RQ2。哪一种特征提取方法和分类模型产生的需求分类结果最精确? 特征提取方法与分类算法的选择对需求分类结果有较大影响。RQ2 旨在选取效果较好的特征提取方

表 6 术语推荐数据集的典型需求

Table 6 Typical requirements for term recommendation datasets

序号	术语推荐数据集的典型需求
1	导线必须采用穿管敷设,电缆应采取防止油、水浸泡和电化腐蚀的措施
2	动力系统的点火装置应有电气和机械联锁的保险机构,防止意外点火
3	发动机除喷管外,任何部位都不允许喷火或冒燃气
4	通用化发控系统的发控单元应接收数字数据总线任何接口处的输入信号波形
5	通用化发控系统的激励电流可以是连续电流或周期性脉冲电流

(2)安全关键软件需求分类方法的需求文本数据集。收集功能、性能、安全性、可靠性和接口 5 类需求各 250 条。数据集涵盖大部分领域的国家军用标准文献,包括军用软件设计要求、航天型号软件设计要求、安全关键软件分析与指南等。安全关键软件需求分类方法的案例规模如表 7 所列。表 8 列出了安全关键软件领域的各类别的典型需求。

表 7 安全关键软件需求分类方法案例规模

Table 7 Case size of safety critical software requirements classification method

需求类别	部分国家军用标准文献	数量
功能	姿轨控系统	250
	卫星对运载火箭的技术要求	
性能	军用直升机飞行性能规范	250
	武器系统性能评定方法	
安全性	航天产品安全性保证要求	250
	军用软件安全性设计指南	
	安全性关键软件分析和开发指南	
	航天型号软件可靠性与安全性设计要求	
可靠性	航天产品可靠性保证要求	250
	航天计算机可靠性设计规范	
	军用软件可靠性评估指南	
	卫星可靠性设计指南	
接口	军用软件接口设计要求	250
	通信系统与保密模块接口要求	
	地面测试系统接口要求	
	炮兵指挥信息系统接口要求	

法与分类模型的可选组合。

### 4.2 实验数据集

本文所使用的实验数据包含两个部分。

(1)安全关键软件术语推荐方法的需求数据集。以航天发射控制子系统需求文档以及国家军用标准等文献中关于航天发射控制相关的文本为术语推荐方法数据集,共选取 234 条需求。案例规模如表 5 所列。

表 5 案例规模  
Table 5 Case size

模块名称	需求语句数量	基准术语数量	基准聚类数量
航天发射控制	234	449	135

表 6 列出了术语推荐数据集中的典型需求。

表 8 安全关键软件各类别的典型需求

Table 8 Typical requirements of various categories of safety-critical software

需求类别	典型需求
功能	整流罩抛罩时,不得有碎片撞击卫星,应采用无污染火工装置
性能	平飞高度-速度范围给出不同重量在不同高度上的最大和最小平飞速度
	最大应急功率应是发动机至少能在 2.5 min 内持续工作的一种功率最大的工作状态
安全性	运载火箭的抛弃级或中止飞行的各级应予回收,或者在其落地前的降落期间安全地摧毁 软件应需要 3 个独立的操作员操作,以激活或者终止一个可能导致一场灾难的危险
可靠性	应对产品进行贮存寿命和贮存可靠性预计,可靠性指标分配应从方案阶段开始,并将可靠性指标分配至发动机各个部组件,再逐步落实到各故障模式
接口	E1 接口用于连接指挥信息系统高速有线电话同传设备和生成多路有线远传信道,K 接口用于数字终端用户与通信网络控制设备、交换机的互连

### 4.3 评估指标

RQ2 使用基础的分类指标来评估需求分类方法,需要注意的是,本文的需求分类方法为多分类问题,因此本文使用 macro 指标,具体的计算方法如式(2)一式(4)所示:

$$P_{macro} = \frac{1}{|G|} \sum_{i=1}^{|G|} \frac{TP_i}{TP_i + FP_i} = \frac{\sum_{i=1}^{|G|} P_i}{|G|} \quad (2)$$

$$R_{macro} = \frac{1}{|G|} \sum_{i=1}^{|G|} \frac{TP_i}{TP_i + FN_i} = \frac{\sum_{i=1}^{|G|} R_i}{|G|} \quad (3)$$

$$F1_{macro} = \frac{1}{|G|} \sum_{i=1}^{|G|} \frac{2 \times P_i \times R_i}{P_i + R_i} = \frac{\sum_{i=1}^{|G|} F1_i}{|G|} \quad (4)$$

其中,  $G$  为类别的集合;  $|G|$  为类别的数量;  $P_i$  与  $R_i$  分别为类别为  $i$  的精度 (precision) 指标和召回率 (recall) 指标。TP (True Positive)、FP (False Positive)、TN (True Negative) 和 FN (False Negative) 分别为真阳性、假阳性、真阴性和假阴性。为了更好地评价分类器的性能, 一般使用 F1-measrue 作为评价标准来衡量分类器的综合性能。

#### 4.4 RQ1: 术语推荐方法

为了评估方法与工具的实用性, 我们设计了人工参与的对照实验, 参与对象为 12 位学生 (6 名本科生和 6 名研究生) 和 6 位工程师。实验步骤如下: 首先, 对所有参与者进行共计 30 min 的 TRRC4SCSTool 工具的使用讲解培训。在后续过程中, 参与者遇到问题可以提问; 然后, 给定两个工业案例 (卫星姿态轨道控制子系统、雷达信息处理子系统) 的需求, 要求

参与者完成构建领域术语表的任务; 最后, 如表 9 所列, 对参与者的任务用时进行统计, 并收集参与对象的反馈意见。实验结果表明:

(1) 手工提取领域术语的平均用时为 1h48min, 远远大于使用本文术语推荐方法的平均时间 (仅使用术语提取为 30m, 使用术语提取和术语聚类为 17m)。

(2) 在所有案例中, 工程师的用时均少于学生的用时, 反映出具备更多专业知识的参与者可以更快地掌握术语推荐方法。

(3) 根据参与者的反馈, 手工完成领域术语对于所有参与者来说均需要花费大量时间, 而采用本文方法可以大大缩短领域术语构建的时间, 具有学习成本低、实用性强的特点, 对于实际案例具有较为积极的意义。

表 9 案例执行时间统计

Table 9 Execution time of case studies

案例	参与对象	方法	执行时间
案例 A 姿态轨道控制系统, 420 条需求, 785 条术语	本科生	手工提取	3 h10 min
		术语推荐 (仅术语提取)	0 h50 min
		术语推荐 (术语提取 + 术语聚类)	0 h30 min
	研究生	手工提取	2 h30 min
		术语推荐 (仅术语提取)	0 h40 min
		术语推荐 (术语提取 + 术语聚类)	0 h25 min
	工程师	手工提取	1 h30 min
		术语推荐 (仅术语提取)	0 h30 min
		术语推荐 (术语提取 + 术语聚类)	0 h15 min
案例 B 雷达系统, 120 条需求, 334 条术语	本科生	手工提取	1 h40 min
		术语推荐 (仅术语提取)	0 h25 min
		术语推荐 (术语提取 + 术语聚类)	0 h15 min
	研究生	手工提取	1 h15 min
		术语推荐 (仅术语提取)	0 h22 min
		术语推荐 (术语提取 + 术语聚类)	0 h10 min
	工程师	手工提取	0 h45 min
		术语推荐 (仅术语提取)	0 h10 min
		术语推荐 (术语提取 + 术语聚类)	0 h5 min

#### 4.5 RQ2: 需求分类方法

##### 4.5.1 实验设计

本文组合不同的特征提取方法和分类算法, 用于将软件需求分为 2 类 (功能需求和非功能需求)、4 类 (性能、接口、可靠性、安全性这 4 类非功能需求) 和 5 类 (功能需求和 4 类非功能需求)。具体而言, 我们考虑 3 种特征提取方法和 5 种分类算法, 共 15 种组合策略。实验基于 Scikit-learn 机器学习库和五折交叉验证的方法进行。具体而言, 在逻辑回归算法方面, 由于 liblinear 在小规模数据环境下收敛速度更快, 因此逻辑回归损失函数的优化方法选择 liblinear。支持向量机算法使用 LinearSVC 函数, 原因是本文的数据集的样本数远小

于特征维数, 若选用非线性核, 如高斯核, 可能产生过学习的问题。朴素贝叶斯算法选取多项式模型贝叶斯分类器, 其中拉普拉斯平滑系数设置为 1。K 近邻算法选用 KNeighbors Classifier 类, K 值设置为 5。随机森林算法采用 RandomForest Classifier 类, 最大弱学习器个数设置为 200, 决策树的最大深度设置为 3, 卡方选择的特征数设置为 500。

##### 4.5.2 实验结果与讨论

表 10 列出了二分类的实验结果, 结果表明: SVM 算法在 3 种特征提取方法中展示了稳定的效果, 且指标均大于 93.75%。LR 算法和 NB 算法在 Bow 和 CHI 上表现良好, 但是与 TF-IDF 组合时, 表现欠佳。

表 10 功能需求和非功能需求二分类

Table 10 Functional and non-functional requirement classification

(单位: %)

	TF-IDF			Bow			CHI		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
LR	95.96	82.40	87.15	97.79	92.50	94.78	98.27	92.80	95.21
NB	94.89	77.20	82.42	98.20	94.30	96.08	98.41	94.35	96.20
SVM	98.88	95.40	96.99	97.17	94.80	95.87	96.66	93.75	95.10
KNN	96.74	93.40	94.93	94.33	74.40	79.67	93.90	77.45	82.38
RF	70.13	50.80	46.08	70.19	51.20	46.86	93.04	67.60	72.10

表 11 列出了四分类的实验结果,结果表明:LR, NB 和 SVM 在 3 种特征提取方法中都表现良好,其中 SVM 与 TF-IDF 组合表现最佳,评价指标均大于 95.50%。

表 12 列出了五分类的实验结果。实验结果表明:SVM 与 TF-IDF 组合同样具有最佳的表现,评价指标均大于 95.59%。

表 11 4 类非功能需求分类  
Table 11 Four non-functional requirements classification

(单位:%)

	TF-IDF			Bow			CHI		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
LR	97.43	97.40	97.40	95.97	95.90	95.90	95.50	95.40	95.40
NB	97.37	97.30	97.30	97.38	97.30	97.30	96.36	96.20	96.20
SVM	97.52	97.50	97.50	95.46	95.40	95.40	95.29	95.20	95.50
KNN	94.62	94.50	94.50	82.29	80.60	80.08	83.48	82.10	81.84
RF	91.83	91.50	91.47	92.97	92.60	92.59	90.79	90.50	90.50

表 12 功能需求和 4 类非功能需求  
Table 12 Functional and four non-functional requirements classification

(单位:%)

	TF-IDF			Bow			CHI		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
LR	97.24	97.20	97.20	94.63	94.56	94.56	93.11	93.01	93.02
NB	96.01	95.84	95.81	95.96	95.76	95.74	94.00	93.60	93.61
SVM	97.64	97.60	97.59	94.57	94.48	94.48	93.12	92.96	92.95
KNN	93.51	93.36	93.34	78.61	75.76	75.38	80.39	78.64	78.17
RF	90.13	89.92	89.91	90.56	90.32	90.32	89.83	89.12	89.18

为了评估 SVM 和 TF-IDF 组合的效果,我们分析该算法在每个类别上的表现,如表 13 所列。实验表明:“功能”类别获得最大的 F1 值,为 98.99%,可靠性类别获得最差的 F1 值,为 91.07%。本文认为导致以上结果的原因可能是可靠性类别过于依赖于关键词的出现,如“可靠性”和“失效率”。

文的时间复杂度和空间复杂度都为立方阶,因此可以较好地满足实际需求。

表 13 TF-IDF+SVM 分类器的效果

Table 13 Classification using TF-IDF+SVM  
(单位:%)

	Precision	Recall	F1
功能	100.00	98.00	98.99
性能	97.47	98.72	98.09
安全性	95.38	98.41	96.88
可靠性	89.47	92.73	91.07
接口	98.41	92.54	95.38

表 14 术语推荐工具对比

Table 14 Comparison with existing terminology recommendation tools

	文献 [20]	文献 [8]	文献 [14]	文献 [21]	本文
适用语言	中文	英文	中文	英文	中文
适用领域	通用	安全关键	通用	通用	安全关键
空间复杂度	$O(n^3)$	$O(n^3)$	$O(n^3)$	$O(n^3)$	$O(n^3)$
时间复杂度	$O(n^3)$	$O(n^2)$	$O(n^3)$	$O(n^2)$	$O(n^3)$
	词性规则	是	是	是	是
提取候选术语规则	依存规则	否	否	是	否
	是否聚类	否	是	否	否
过滤候选术语规则	常用词过滤	是	是	是	是
	领域度过滤	否	否	否	是

表 15 需求分类工具对比

Table 15 Comparison with existing requirement classification tools

	文献 [22]	文献 [6]	文献 [12]	本文
适用语言	英文	英文	中文	中文
适用领域	通用	通用	通用	安全关键
空间复杂度	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(n^2)$
时间复杂度	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(n^2)$
需求类别	5 类	12 类	5 类	5 类
算法	机器学习	机器学习	加权投票	机器学习

由于本文的实验数据集较全面地涵盖了安全关键领域的需求特征,并且数据集的划分过程采用随机抽取的方法,因此实验结果具有一定的通用性。而表 9—表 11 中列出的单次实验结果也能够一定程度上体现不同组合策略的稳定性。综上所述,对于安全关键软件的需求描述,上述组合策略取得了良好的测试结果。本文的实验结果更适合各种需求分散在整个安全关键软件需求文档中的应用场景。例如,用户可以使用未处理的需求文档作为输入,然后使用本文推荐的组合策略来识别需求文本的可能类别,从而降低手动检查的范围。

#### 4.6 比较

表 15 列出了需求分类工具与部分现有工具的对比,如文献 [6, 12, 22] 等提出的工具。从表中可以看出,本文方法支持中文安全关键软件领域的需求分类。Jia 等 [12] 的工作采用加权投票的算法来对需求进行分类,其余大部分工作采用机器

我们将本文所提术语推荐工具与现有的一些工具进行对比,如文献 [8, 14, 20-21] 等提出的工具,主要包括适用语言和领域、算法复杂度、提取规则和过滤规则等方面的比较,如表 14 所列。从表中可以看出,本文方法能支持中文安全关键软件领域的术语推荐,对基于依存规则的提取术语、领域度过滤、候选术语聚类有较为完整的支持。在算法复杂度方面,本

学习的方法。但是,相比 Lu 等<sup>[22]</sup>、Canedo 等<sup>[6]</sup>的研究,本文组合不同的特征提取方法和分类算法,即 3 种特征提取方法和 5 种分类算法,共 15 种组合策略(见 4.5 节),从而可以比较不同特征提取方法和分类方法的使用效果。

#### 4.7 有效性威胁

##### (1) 内部有效性威胁

首先,基准术语结果与基准术语聚类结果都是主观的过程。为了减少这一威胁,本文采用先由两名研究生进行标注,再由领域专家进行审查的方式来决定基准数据。

其次,本文使用人工标记的样本数据进行安全关键软件需求分类研究,因此样本质量是影响分类方法的内部有效性威胁之一。为了减轻这一威胁,我们详细参阅了国家军用标准中《军用软件需求分析》等文献,目的是确定典型的需求类别,并且保证研究者以相同的方式理解各类安全关键软件需求。数据集收集过程由两名研究生和一名领域专家共同完成。这些措施可以缓解数据集收集过程中的个人偏见,从而减轻人工标记对分类方法的内部有效性威胁。

最后,领域专家的审查结果可能有助于本文方法的实验结果。为了减轻这一威胁,在审查过程前后,领域专家并不被告知本文方法。

##### (2) 外部有效性威胁

首先,实验数据集并不涉及数据、环境、可修改性以及假设和约束等需求类别。因此,收集更全面、丰富的安全关键软件需求数据集有助于减轻这一外部有效性威胁。其次,本文的数据集涵盖大部分领域的国家军用标准文献,包括军用软件设计要求、航天型号软件设计要求、安全关键软件分析与指南等。然而,更多的工业界案例研究对提高外部有效性是十分必要的。

## 5 相关工作

### 5.1 术语自动提取

自动术语提取研究可分为 3 种方式:语言学、统计和混合<sup>[7]</sup>。

#### (1) 语言学方法

根据语言特性来制定提取术语的规则模板,其中规则模板主要根据语法、词法和句法等语言学规则和领域特征构建而成。Earl 等<sup>[23]</sup>率先提出术语一般为名词,之后的很多研究都在此基础上完成;Bourigault<sup>[24]</sup>首先通过消除代词和限定词之类的停用词,然后使用基于词性的正则表达式来提取名词短语;Aubin 等<sup>[25]</sup>使用分块(chunking)和句法解析的组合来提取简单和复杂的名词短语;Stanković 等<sup>[26]</sup>依靠广泛的电子词典和有限状态转换器形式的词汇资源,构造出多词术语的句法结构模板,并通过模板来提取候选术语;Arora 等<sup>[8]</sup>利用词性规则模板匹配出候选术语表,并对候选术语表进行聚类,最终通过真实工业界案例进行评价,产生了较好的效果。

#### (2) 统计方法

根据统计指标(如频率和长度)选择术语。Jones 等<sup>[27]</sup>通过给文档中的单词序列分配等级的方式来识别关键字;Berry 等<sup>[28]</sup>和 Goldin 等<sup>[29]</sup>提出了基于频率的方法来识别在需求中反复出现的术语;Matsuo 等<sup>[30]</sup>使用单词和单词序列的共现

频率来识别关键术语。

#### (3) 混合方法

针对语言学和统计两种方法的不同特点,两种方法混合使用成为重要选择。Frantzi 等<sup>[31]</sup>提出 C-value/NC-value 方法,首先基于语言学方法抽取名词短语,之后利用词频和词长来抽取复合术语;Barker 等<sup>[32]</sup>首先使用文本分块来识别给定文本中的名词词组,然后根据频率和长度过滤掉不太可能的术语;Park 等<sup>[33]</sup>使用词性模板提取候选术语,然后根据领域相关度来过滤术语;Zou 等<sup>[34]</sup>使用词性模板提取单词和双词的名词短语,然后根据频率和某些启发式方法过滤结果;Dwarakanath 等<sup>[35]</sup>使用句法解析来提取需求文档的短语,然后基于启发式方法和基于频率的统计信息过滤结果;Gemkow 等<sup>[36]</sup>首先通过语言学处理来识别候选术语,然后通过统计学方法来增加术语相关性和领域特异性。

### 5.2 术语聚类

术语聚类方法分为两种类别:基于模式的方法和基于语法和语义相似度的方法。

#### (1) 基于模式的方法

依赖于预定义的词性模式来识别术语相关性。Daille<sup>[37]</sup>使用前缀词性标签来识别形容词和介词修饰。Bourigault 等<sup>[38]</sup>根据名词修饰语的模式对相关术语进行分组。

#### (2) 基于语法和语义相似度的方法

通过语法和语义相似性度量来检测术语相似度。Arora 等<sup>[8]</sup>首先计算术语的语法及语义相似度并生成相似矩阵,然后使用聚类算法来对术语进行聚类。

### 5.3 需求分类

Cleland<sup>[39]</sup>等提出了一种基于特征词的非功能需求识别方法,从人工标注的训练数据中提取各类非功能需求的特征词,并根据特征词的出现频率对文本进行分类;Hindle 等<sup>[40]</sup>采用了不同的思路,直接基于话题模型获得词汇列表,进而将其对应到各类非功能需求;Sunner 等<sup>[10]</sup>使用基于遗传算法的神经网络,实现了非功能需求和功能需求的分类;Lu 等<sup>[22]</sup>将 4 种分类技术和 3 种机器学习方法相结合,可以将用户评论自动分类为 4 种类型的非功能需求(可靠性、可用性、可移植性和性能)、功能需求和其他。

由于语言之间的巨大差异,这些研究成果无法直接为中文需求文本提供支持。中文具有词语间无空格分隔、歧义现象普遍、实词运用灵活等特点,更是形成了其特有的复杂性和难度。针对中文需求文本,Jia 等<sup>[12]</sup>提出了一种自动化的非功能需求识别和分类方法,其中分类过程则基于语义距离和相似度计算,将包含非功能需求语句划分为性能、可靠性、可用性、安全性、可维护性 5 类。

综上所述,在自动术语提取、术语聚类和需求分类方面,已有研究主要关注通用软件领域,安全关键软件领域的研究相对较少,且较少针对中文文本。

**结束语** 本文提出了一种安全关键软件术语推荐和需求分类方法,为安全关键软件需求规约提供了基础。在术语推荐方面,将安全关键软件需求、安全分析与认证标准、安全分析报告等作为输入,经过候选术语提取、术语相似度计算和候选术语聚类 3 个步骤的处理,获得候选术语表。在需求分类

方面,针对输入数据集,经过文本预处理、提取文本特征、训练分类器和评估分类器4个步骤处理后,得到分类器,并将安全关键软件需求输入分类器,进而得到可能的需求类别。同时,设计和实现了 TRRC4SCSTool 原型工具,并将工业界实际案例需求、安全分析与认证标准、安全分析报告等作为实验数据集,验证了本文方法的可用性和有效性。

下一步研究主要包括:首先,探索将用户反馈加入到术语推荐方法中,以便进行更好的推荐。如让工程师构建一个初步的领域术语库来辅助分词工作,有望优化术语推荐的质量;其次,在需求分类方法中,本文主要采用传统机器学习方法,我们将进一步考虑使用深度学习;最后,机器学习支撑的系统应用越来越普遍,但是此类系统的需求通常难以表达完整<sup>[41]</sup>。因此,我们正在研究带 AI 构件的安全关键软件需求术语推荐与分类方法。

### 参 考 文 献

[1] LEVESON N G. Engineering a safer world: Systems thinking applied to safety [M]. Cambridge: The MIT Press, 2016: 7-15.

[2] RTCA DO-178C. Software Considerations in Airborne Systems and Equipment Certification[S]. Washington, DC: RTCA, 2011.

[3] VILELA J, CASTRO J, MARTINS L E G, et al. Integration between requirements engineering and safety analysis: A systematic literature review[J]. Journal of Systems and Software, 2017, 125: 68-92.

[4] YUE T, BRIAND L C, LABICHE Y. Facilitating the transition from use case models to analysis models: Approach and experiments[J]. ACM Transactions on Software Engineering and Methodology (TOSEM), 2013, 22(1): 1-38.

[5] WANG F, YANG Z B, HUANG Z Q, et al. An Approach to Generate the Traceability Between Restricted Natural Language Requirements and AADL Models [J]. IEEE Trans. Reliability, 2020, 69(1): 154-173.

[6] CANEDO E D, CORDEIRO M B. Software Requirements Classification Using Machine Learning Algorithms [J]. Entropy, 2020, 22(9): 1057.

[7] PAZIENZA M T, PENNACCHIOTTI M, ZANZOTTO F M. Terminology extraction: an analysis of linguistic and statistical approaches[M]. Knowledge Mining. Berlin, Heidelberg: Springer, 2005: 255-279.

[8] ARORA C, SABETZADEH M, BRIAND L, et al. Automated Extraction and Clustering of Requirements Glossary Terms[J]. IEEE Transactions on Software Engineering, 2017, 43(10): 918-945.

[9] KURTANOVIC Z, MAALEJ W. Automatically classifying functional and non-functional requirements using supervised machine learning[C]// Proceedings of the 2017 IEEE 25th International Requirements Engineering Conference (RE). Lisbon, Portugal, 2017: 490-495.

[10] SUNNER D, BAJAJ H. Classification of Functional and Non-functional Re-quirements in Agile by Cluster Neuro-Genetic Approach[J]. International Journal of Software Engineering and Its Applications, 2016, 10(10): 129-138.

[11] ABUALHAIJA S, ARORA C, SABETZADEH M, et al. Automated demarcation of requirements in textual specifications: a machine learning-based approach[J]. Empirical Software Engineering, 2020, 25(6): 1-44.

[12] JIA Y, LIU L. Recognition and Classification of Non-functional Re-quirements in Chinese [J]. Journal of Software, 2019, 30(10): 3115-3126.

[13] HE H. HanLP: Han Language Processing[OL]. [2020-6-1]. <https://github.com/hankcs/HanLP>.

[14] YU Y, CHEN L, JIANG J, et al. Research on Chinese Patent Candidate Terminology Selection Based on Dependency Syntax Analysis[J]. Library and Information Work, 2019, 63(18): 109-118.

[15] GOMAA W H, FAHMY A A. A survey of text similarity approaches[J]. International Journal of Computer Applications, 2013, 68(13): 13-18.

[16] MANNING C D, RAGHAVAN P, SCHÜTZE H. Introduction to information retrieval[M]. Cambridge: Cambridge University Press, 2008.

[17] COHEN W W, RAVIKUMAR P, FIENBERG S E. A Comparison of String Distance Metrics for Name-Matching Tasks[C]// IWeb Conference. UAC: ACM2003, 2003: 73-78.

[18] SOOD S, LOGUINOV D. Probabilistic near-duplicate detection using simhash [C] // Proceedings of Proceedings of the 20th ACM international conference on Information and Knowledge Management. USA: ACM, 2011: 1117-1126.

[19] FEILER P. The Open Source AADL Tool Environment (OS-ATE) [R]. Pitts-burgh: Carnegie Mellon University Software Engineering Institute, 2019.

[20] ZENG Z, LV X, LI Z, et al. A patent abstract-oriented domain term extraction method [J]. Computer Application and Software, 2016, 33(3): 48-51.

[21] MISHRA S, SHARMA A. Automatic Word Embeddings-Based Glossary Term Extraction from Large-Sized Software Requirements [C]// International Working Conference on Requirements Engineering: Foundation for Software Quality. Springer, Cham, 2020: 203-218.

[22] LU M M, LIANG P. Automatic classification of non-functional re-quirements from augmented app user reviews [C] // Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering. 2017: 344-353.

[23] EARL L L. Experiments in automatic extracting and indexing [J]. Information Storage and Retrieval, 1970, 6(4): 313-330.

[24] BOURIGAULT D. Surface grammatical analysis for the extraction of terminological noun phrases [C] // Proceedings of the 14th conference on Computational linguistics-Volume 3. UAS: Association for Computational Linguistics, 1992: 977-981.

[25] AUBIN S, HAMON T. Improving term extraction with terminological resources [C] // Proceedings of International Conference on Natural Language Processing (in Finland). Germany: Springer, 2006: 380-387.

[26] STANKOVIĆ R, KRSTEV C, OBRADOVIĆ I, et al. Rule-based automatic multi-word term extraction and lemmatization [C] // Proceedings of Proceedings of the Tenth International Confe-

- rence on Language Resources and Evaluation (LREC'16). France: European Language Resources Association, 2016: 507-514.
- [27] JONES L P, GASSIE E W, RADHAKRISHNAN S. INDEX: The statistical basis for an automatic conceptual phrase-indexing system[J]. *Journal of the American Society for Information Science*, 1990, 41(2): 87-97.
- [28] BERRY D M. The use of a repeated phrase finder in requirements extraction[J]. *Journal of Systems and Software*, 1990, 13: 209-230.
- [29] GOLDIN L, BERRY D M. AbstFinder, a prototype natural language text abstraction finder for use in requirements elicitation [J]. *Automated Software Engineering*, 1997, 4(4): 375-412.
- [30] MATSUO Y, ISHIZUKA M. Keyword extraction from a single document using word co-occurrence statistical information[J]. *International Journal on Artificial Intelligence Tools*, 2004, 13(1): 157-169.
- [31] FRANTZI K, ANANIADOU S, MIMA H. Automatic recognition of multi-word terms; the c-value/nc-value method[J]. *International Journal on Digital Libraries*, 2000, 3(2): 115-130.
- [32] BARKER K, CORNACCHIA N. Using noun phrase heads to extract document keyphrases[C]// *Proceedings of Conference of the Canadian Society for Computational Studies of Intelligence*. Germany: Springer, 2000: 40-52.
- [33] PARK Y, BYRD R J, BOGURAEV B. Automatic glossary extraction; beyond terminology identification[C]// *Proceedings of COLING 2002: The 19th International Conference on Computational Linguistics*. Spain: COLING, 2002.
- [34] ZOU X, SETTIMI R, CLELAND-HUANG J. Improving automated requirements trace retrieval; a study of term-based enhancement methods[J]. *Empirical Software Engineering*, 2010, 15(2): 119-146.
- [35] DWARAKANATH A, RAMNANI R R, SENGUPTA S. Automatic extraction of glossary terms from natural language requirements[C]// *Proceedings of 2013 21st IEEE International Requirements Engineering Conference (RE)*. USA: IEEE, 2013: 314-319.
- [36] GEMKOW T, CONZELMANN M, HARTIG K, et al. Automatic glossary term extraction from large-scale requirements specifications[C]// *Proceedings of 2018 IEEE 26th International Requirements Engineering Conference (RE)*. UAS: IEEE, 2018: 412-417.
- [37] DAILLE B. Variations and application-oriented terminology engineering[J]. *Terminology*, 2005, 11(1): 181-197.
- [38] BOURIGAULT D, JACQUEMIN C. Term extraction+term clustering: An integrated platform for computer-aided terminology [C] // *Proceedings of Ninth Conference of the European Chapter of the Association for Computational Linguistics*. European: EACL, 1999.
- [39] CLELAND-HUANG J, SETTIMI R, ZOU X, et al. Automated classification of non-functional requirements[J]. *Requirements Engineering*, 2007, 12(2): 103-120.
- [40] HINDLE A, ERNST N A, GODFREY M W, et al. Automated topic naming to support cross-project analysis of software maintenance activities[C]// *Proceedings of Proceedings of the 8th Working Conference on Mining Software Repositories*. 2011: 163-172.
- [41] YANG L, MA J, JIANG H, et al. Requirements Modeling and Decision-making for Machine Learning Systems[J]. *Computer Science*, 2020, 47(12): 42-49.



**YANG Zhi-bin**, born in 1982, Ph.D, associate professor, is a member of China Computer Federation. His main research interests include safety-critical systems, formal verification and AI software engineering.