针对需求缺陷检测任务的自然语言需求数据集评估

蔡一涵 马立鹏 杨卫东 施伯乐
(复旦大学计算机科学与技术学院 上海 200438)

摘 要 自然语言是软件需求的主要书写形式之一,易于理解但容易产生缺陷。目前,基于自然语言处理等技术解决需求缺陷的方法引起学术界和工业界的广泛关注。但不像其他领域中存在大量可用公开数据集,在软件工程领域,仍然缺乏合适数据集与评价数据集是否合适的方法来帮助进行基于自然语言的需求缺陷检测等任务。针对需求缺陷检测,提出对应的数据集评估方法与度量模型,设计基于规则的数据集评估框架,对已有的公开需求数据集进行实验分析,并根据量化指标进行统计。

关键词 软件需求 需求缺陷 需求工程 自然语言处理

中图分类号 TP3 **文献标志码** A DOI:10.3969/j.issn.1000-386x.2024.11.011

NATURAL LANGUAGE REQUIREMENT DATASET EVALUATION FOR REQUIREMENT DEFECT DETECTION TASK

Cai Yihan Ma Lipeng Yang Weidong Shi Bole
(*School of Computer Science, Fudan University, Shanghai 200438, China*)

Abstract Natural language has been widely used as one form of software requirements as it is easy to understand. But natural language requirements are prone to defects. At present, applying natural language processing techniques on requirement defects has gradually become a research hotspot. However, unlike other fields having a large number of publicly available datasets, in the field of software engineering, there is still a lack of suitable datasets and methods to evaluate whether datasets are sufficient for helping perform tasks such as natural language defect detection. Aiming at the task of requirement defect detection, we propose an evaluation method and quantitative metric model for corresponding dataset, and designe a rule-based evaluation framework. We experimented with existing public requirement dataset, and conducted statistics based on quantitative metrics.

Keywords Software requirement Requirement defect Requirement engineering Natural language processing

0 引 言

需求通常由自然语言写成,并被认为是开发人员和客户之间进行交流的辅助工具。用自然语言书写而成的需求是一组提供了有关包含在系统中的所需功能信息的语句的集合,它自由、灵活、易于理解,是开发人员最常用的用于与软件使用者进行沟通的工具,但它也因为灵活性和自由度存在容易携带缺陷的问题。作为交流方式中最灵活的类型之一,自然语言有其本身天然的缺陷——例如歧义性、不正确性、非原子性等,而这也是被自然语言需求所继承的。这些缺陷可能导致软件开发的后续阶段存在严重的问题[9],开发人员也期望能够在开发过程的早期阶段检测到自然语言需求的缺陷,因此,需求缺陷检测已经和需求分类、需求摘要抽取等任务一样,成为需求工程领域内采用智能化辅助手段协助提升自然语言需求质量的新的研究热点。已有的工作已提出了许多方法和框架来帮助开发人员检测需求缺陷。

目前已经有研究者为检测需求缺陷的研究工作

收稿日期:2021-07-07。国家自然科学基金项目(U2033209)。**蔡一涵**,硕士生,主研领域:自然语言需求。**马立鹏**,博士生。**杨卫东**,教授。**施伯乐**,教授。

提出了公开的需求数据集,但是,这样的数据集还很少,而且缺少针对此类公开数据集的评判标准,同时这些数据集也需要训练模型、验证实验所需要的标注。

本文针对需求缺陷检测这一任务,对现有的公开需求数据集[1]能否满足此项任务进行了评估,包括需求缺陷的种类、覆盖程度,一方面提出了评估自然语言需求缺陷类型的框架与算法,另一方面补充了现有数据集在缺陷类型标注上的缺失。

1　相关研究

需求工程领域和其他计算机学科内有着大量公开数据集的领域不同,仍然缺乏合适的需求数据集以及评价数据集是否合适的方法。本节一方面将从直接与数据集挂钩相关的工作展开介绍,包括自然语言、计算机视觉等领域对数据集进行评估的工作,与软件工程领域内目前构建公开需求数据集的工作;一方面将介绍提高需求质量的工作,这些研究工作使用到的需求数据说明了公开数据集的重要性、稀缺性,以及提出针对数据集的评估框架的意义。

1.1　数据集相关工作

虽然目前需求工程领域内缺少对数据集进行评估的框架,但在其他领域内已有许多研究工作围绕数据集评估展开:Oh[22]针对分类问题提出了从分类重叠角度评估数据集的方法。Röder 等[23]针对命名实体识别与消歧问题评估了已有的数据集,并经由人工标注、整合提出了一个全新的语料库。Sharafaldin 等[21]针对常见的四个入侵检测数据集进行了评估,并提出了新的公开数据集。Sharghi 等[24]围绕视频摘要查询提出了对应的评估指标,并标注、编译了新数据集。可以看出,数据集的量化评估对于围绕研究问题提高数据集质量非常重要。

在自然语言需求处理方面,为了将自然语言处理技术应用于需求工程,收集可用作基准的需求文档以训练模型和评估算法是非常重要的工作。但是,需求领域的公开需求数据集并不如其他领域那么多。Ferrari 等[1]从互联网上收集了 79 篇公开可用的自然语言需求文档,并将其整合在 PURE(PUblic REquirements)数据集中。该数据集被设想应用于需求工程中典型的自然语言处理任务,他们将数据集文本与通用英语文本进行了比较,并手动将一部分文档转换成了通用 XML 格式。Alhoshan 等[2]开发了 FN-RE,一个基于 FrameNet 语义框架注释的需求文档集,他们从互联网上收集了 34 篇需求文档,然后通过选择合适的语义框架和相关的框架元素,手动标记每句需求语句,并将输出结果统一为 CSV 格式。

在评估已有的公开自然语言需求数据集时,本文选择采用 PURE 数据集作为评估对象,原因有:(1) FN-RE 需求数据集的主体部分直接复现了 PURE 数据集的构建过程与 XML 格式规范;(2) FN-RE 数据集在规模上远小于 PURE 数据集,且 FN-RE 数据目前的规模不足以支持大规模的模型学习训练;(3) FN-RE 数据集的标注部分为格式化需求数据,与本文讨论的自然语言需求数据集不符。

1.2　提升需求质量的工作

与指定格式的结构化需求文档相比,自然语言需求往往更易于理解和使用,也因此导致对其提高需求质量的工作相对于结构化需求难度更高。基于自然语言处理等技术解决需求缺陷的方法引起学术界和工业界的广泛关注,但用于相关研究的公开数据集严重不足。

在进行需求缺陷检测、协助提高需求文档质量的文献工作中,最著名的工作可能是美国国家航空航天局(NASA)提出的 ARM(Automated Requirements Measurement)工具[3-4]。目前常见的需求缺陷检测工作主要分为两类,一种基于给定的需求文档模板或者需求语句规范,另一种针对给定的需求缺陷类型(或者是特定的某种缺陷)进行检测。

前一类基于给定的需求规范模板的缺陷检测工作包括:Arora 等[5]提出了 RUBRIC(ReqUirement BoileRplate sanIty Checker),以作为一种用于自动检查需求是否符合要求的灵活工具,他们的实验是在 380 句来自工业领域的需求语句上完成的。Huertas 等[6]开发了 NLARE(Natural Language Processing Tool for Automatic Requirements Evaluation),并提出了一些用于需求规范语句结构的准则,他们在一款现实 Wi-Fi 追踪设备的需求文档上测试了这一款工具。Ambriola 等[7]提出了 Circe,一款基于 Web 的、用于协助自然语言需求分析的工具,他们将这一集成工具应用在了学术和工业场景的多个实验中。

后一类针对特定的需求缺陷类型进行检测需求缺陷检测工具有:Rosadini 等[8]使用自然语言处理技术来检测需求缺陷,将其应用于铁路领域的工业需求实例中,并将传统的手动分析与使用自然语言处理技术的需求分析进行了比较。Lami 等[9]开发了 QuARS

(Quality Analyzer for Requirement Specifications),用于系统地解析自然语言需求,并辅助需求工程师对需求中潜在的语言缺陷进行分析。

可以看出,基于规则的缺陷检测模型和以辅助人工分析为主要目的的工具并非一定依赖需求文档数据才能进行;然而对于涉及需要训练集、验证集的模型的工作,目前研究人员在选择实验数据时往往无法选择公开需求数据集,只能在工业的、企业的、私有的需求文档上先自行标注,再进行实验。这也就导致,相关的实验工作难以复现,模型的效果难以进行统一评判。因此,针对需求缺陷检测提出针对数据集的度量模型与评估框架,对于后续的数据集构建与领域内的研究有着重要意义。

2 度量模型

为了建立对公开需求数据集的评判标准,针对需求缺陷检测,本文提出了评估数据集所必要的度量模型,包括需求数据的形式化定义,针对缺陷类型的量化指标,与对应的缺陷类型定义。

2.1 形式化定义

我们定义,对于每句需求语句 $s(i)$ 和每种需求缺陷类型 De_x,其所属于的需求文档为:

$$Req = \{s(1), s(2), \cdots, s(n)\} \tag{1}$$

有:

$$De_x(s(i)) = \begin{cases} 1 & s(i) \text{ carries this kind of defect} \\ 0 & \text{其他} \end{cases} \tag{2}$$

从而可以将一篇需求文档转化为一系列包含了需求陈述语句的元组的集合 W:

$$S(i) = \{s(i), De_1(s(i)), De_2(s(i)), \cdots\} \tag{3}$$

$$W = \{S(1), S(2), \cdots, S(n)\} \tag{4}$$

2.2 量化指标

基于需求数据的形式化定义,可以对需求数据集的特征进行量化评估,本文提出了以下几项量化指标。

已知有包含 x 类缺陷类型与 n 句需求语句数据的自然语言需求数据集,对于给定的需求缺陷类型 De_a,我们定义其缺陷覆盖率 CR_a:

$$CR_a = \frac{\sum_{i=1}^{n} De_a(s(i))}{n} \tag{5}$$

数据集的覆盖标准差 σ 为:

$$\sigma = \sqrt{\frac{\sum_{a=1}^{x} (CR_a - \overline{CR})^2}{x}} \tag{6}$$

则给定缺陷类型 De_a 的覆盖偏差值 DV_a 为:

$$DV_a = \frac{10 \times (CR_a - \overline{CR})}{\sigma} + 50 \tag{7}$$

若某类缺陷的覆盖偏差值高于 50,说明该缺陷类型的覆盖率在平均之上,反之则偏少,且偏差值偏离基准值 50 的绝对值越大,说明此类缺陷覆盖率偏离平均越多。数据集覆盖偏差的趋势越明显,则数据集的缺陷分布越不均衡。

对于给定的两种缺陷类型 De_a 与 De_b,定义其相对重叠度 $RelativeOL_{ab}$:

$$RelativeOL_{ab} = \frac{\sum_{i=1}^{n} (De_a(s(i)) \times De_b(s(i)))}{n} \tag{8}$$

相对重叠度越高,说明该数据集中这两类需求缺陷同时在需求语句中出现的情况越多。

对于数据集整体,定义其全局重叠度 $GlobalOL$:

$$GlobalOL = \frac{\sum RelativeOL}{x \times (x-1)} \tag{9}$$

重叠度描述了需求数据集在不同缺陷上重叠的特征,全局重叠度越高,则该数据集中的缺陷类型出现情况越复杂;而不同缺陷的相对重叠度高,则说明这些缺陷往往相伴而生,在未来针对不同领域的需求文档数据集内这种特征可能会更加突出。缺陷重叠度越高说明缺陷出现的情况越复合、越复杂,使用简单模型(如基于规则)进行缺陷检测任务越困难,反之重叠度越低的需求数据,特征的提取越明显。

将需求语句 $s(i)$ 对应的缺陷类别标记 $De_x(s(i))$ 视为一个 x 维向量,则可定义多缺陷比重 MDR:

$$\varphi = \sum_{i=1}^{n} \boldsymbol{\Phi}_{De}\{\boldsymbol{I}[\sum_{\alpha=0}^{x} De_x(s(i))]\} \in \mathbf{R}^{1\times x} \tag{10}$$

$$MDR = \frac{\varphi_\alpha}{\sum_{\alpha=0}^{x} \varphi_\alpha} \tag{11}$$

式中:φ_α 表示第 α 个元素为 1,其余元素为 0 的行向量;$\boldsymbol{I}$ 代表指示函数。多缺陷比重代表了具备多种缺陷的需求语句的比重,缺陷数量多的占比越高,数据集中的语句所具有的缺陷越倾向于复合,数据集越适合用于复杂任务。

2.3 缺陷类型定义

根据已有的需求缺陷检测工作,我们提出了作为评判数据集能否满足需求缺陷检测实验的标签类型。

我们划分需求缺陷类型的基准包括:该缺陷类型在已有的研究工作中被提及、涉及过,如对二义性、模糊性的研究等。该缺陷类型在现实自然语言需求语句中经常出现,检测此类缺陷对提升需求质量具有现实应用场景的意义;在针对该自然语言需求缺陷的检测中,现有的技术与模型能够有效且可解释地完成任务;需求缺陷类型可以通过实验证实其有效覆盖。

本文定义的自然语言需求缺陷类型为:模糊性,包括指代模糊性、并列模糊性、模糊副词与短语;非原子性,包括连词导致的非原子性、从句导致的非原子性;不完整性,包括被动语态、条件缺失。

1) 模糊性。

定义:当需求语句中的单词或多词短语存在两个或多个可用含义时,该需求语句就存在模糊性缺陷。

(1) 指代模糊性(Anaphoric Ambiguity)。

如果需求文本在同一语句或上下文的语句中提供了多个以上可用于解释指代词的选项,就会产生指代模糊性[12]。

(2) 并列模糊性(Coordination Ambiguity)。

并列模糊性是由使用并列连词引起的,随意使用此类连词可能导致一句需求语句存在多种可能的解读方式[8]。第一类情况是一句需求语句中存在两个及以上的并列连词,第二种情况是需求语句混合使用了并列连词和修饰语。

(3) 模糊副词与短语(Ambiguous Adverbs and Phrases)。

无法确定需求语句中涉及数量的真值也会带来模糊性,即使用没有精确语义的单词、词组对系统或者构件进行描述。

2) 非原子性。

许多需求设计可以从原子性的概念中受益,因为原子性的单一概念有助于分析隐藏在由多个步骤组成的动作或组件中的软件需求[13]。

(1) 连词导致的非原子性(Non-Atomicity from Conjunction)。

本文定义,当需求语句使用不必要的连词连接两句或更多完整句子时,视为存在连词导致的非原子性。

(2) 从句导致的非原子性(Non-Atomicity from Clause)。

从句的嵌套使用也会导致非原子性缺陷,使用嵌套的从句导致需求语句变得冗长而混乱,理解需求变得困难。

3) 不完整性。

如果一句需求语句缺少动作主语、功能或条件,视为该需求语句不完整[14]。基于此原理,本文定义了两种类型的缺陷,以对应不完整性缺陷与自然语言需求语句中可能缺少的因素。

(1) 被动语态(Passive Voice)。

使用被动语态会导致需求的不完整性缺陷,因为如果动作的执行者没有紧跟在语句中,被动语态就可能带来误解[8]。

(2) 条件缺失(Missing Condition)。

为了确保需求语句的完整性,每个通过 if 从句表达条件的需求语句都应该存在一个 else 或 otherwise 从句进行对应[8]。

3　评估框架

3.1　预处理

根据相关工作中的讨论,我们选择对 PURE 数据集进行有关缺陷检测任务的评估。现有的 PURE 数据集并无自然语言需求缺陷的标注,因此首先需要对 PURE 中的需求文档进行标注,此后才能对标签类型的覆盖结果进行评判。

首先我们对 PURE 数据集中的需求文档建立了纳入/排除标准,为后续的标注工作进行筛选。纳入标准为:在使用自动化工具转换成文本格式后没有丢失大量需求语句;原始文档内格式统一、清晰,转换后没有出现大量拼写错误;描述的系统或系统构件较为完整。排除标准为:在手动方法和目前已有的自动化工具中都难以进行标注,其篇幅过长且文本过于模糊导致无法通过自动化工具获得足够准确的结果;使用完全结构化或者形式化的方式书写,不属于本文讨论的自然语言需求范围。

在具体按照纳入/排除规则进行筛选的过程中,我们首先通过 pdfminer 和 pdftotext 一类的自动化转换库将非纯文本格式的需求文档批量转化成文本格式,在这一过程中,我们也删除了非自然语言的部分(如照片和图形)。之后由 4 名硕士研究生、博士研究生两两分组按照纳入/排除规则对转换所得的纯文本需求进行评价,其中满足纳入标准少于半数或者满足排除标注超过半数的需求文档将被划分入“排除”类,反之则划分入“纳入”类。当同组内的两人意见一致时视为该文档的划分没有分歧,当两人意见不一致时交由另一组的两人再次进行讨论与评估。

3.2 框架设计

在前述定义的基础上,对需求文档进行标注的框架基于 GATE,作为一款开源软件工具包,它可以用作包括自然语言需求处理流程的框架。基于 GATE 框架的标注模型如图 1 所示。

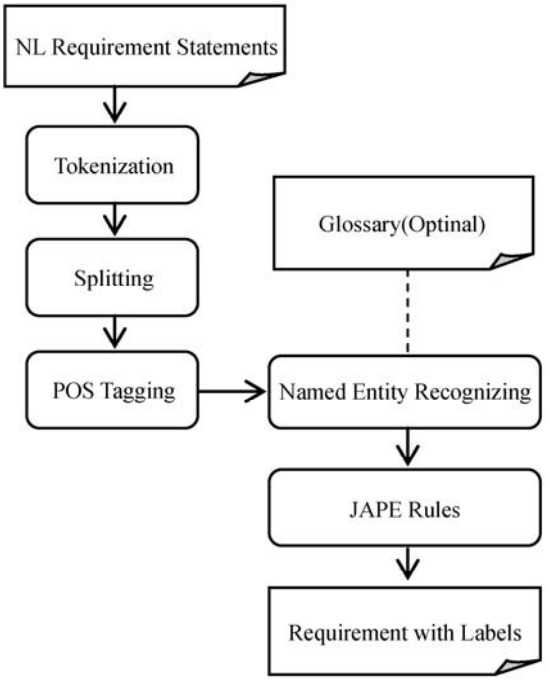

图 1 基于 GATE 对需求文档进行标注

第一步需要将输入的需求文本进行分词(tokenization),分词是一种将输入的文本序列分成称为标识(token)的较小单元的方法。标识可以是单词、字符或子单词。因此,分词可以包括单词分词、字符分词和子词(N-gram 字符)分词等。大多数后续的处理模块都需要使用以这种方式标记的文本。本框架中使用的是 OpenNLP[17] 的分词器,其分词过程分为两个阶段:首先识别句子边界,随后识别每个句子中的标识。

随后使用拆分器(Splitter)将文本一一拆分为句子,并使用词性标记器(Part-of-Speech Tagger)为每个标识添加注释 POS 标记(又称词性标记)。这一步是基于单词的定义和上下文,将文本中的单词标记为与文本词性相对应的过程。这些标签包括形容词、副词、名词、动词等。在拆分句子时,拆分器检测给定的标点符号是否标记了句子的结尾。也就是说,一句句子的定义为两个标点符号之间最长的缩减了空白字符的序列(第一句和最后一句句子例外)。句子的第一个非空白字符被认为是句子的开头,而最后一个非空白字符被认为是句子的结尾。

对于词性标记器,我们选用的是 GATE 中的 Stanford PoS Tagger[15] 组件,它使用的是最大熵模型,最大熵模型的优点在于,在建模时使用者只需要集中精力选择特征,而不需要花费精力考虑如何使用这些特征。利用最大熵建模,一般也不需要做在其他方法建模中常见的独立性假设,参数平滑可以通过特征选择的方式加以考虑,无须专门使用常规平滑算法单独考虑。在英文的词性标注中使用到的是 Penn Treebank 的标注集[16],数据采用《华尔街日报》(WSJ)0 - 18 节进行训练,使用 19 - 20 节进行测试。

下一步是命名实体识别(Named Entity Recognizing),识别文本中具有特定意义的实体,主要包括人名、地名、机构名、专有名词等,以及时间、数量、货币、比例数值等文字。在处理自然语言需求的过程中,这一步的目的往往在于识别组织、域关键字和组件名称,因此可以在该模块中添加术语词汇表作为参考补充。对于英文的自然语言需求来说,人名、地名、组织等实体使用 CRF 模型[18] 来识别,对于日期、时间等则使用基于规则的方法。这里使用的 Stanford NER Tagger[18] 模型是基于 CoNLL[19]、MUC[20] 和其他一些额外的语料库作为数据训练得到的。

最后通过 JAPE(Java Annotation Patterns Engine)脚本编写的规则,模型就可以检测给定的自然语言需求缺陷。下面为一段检测并列模糊性缺陷的 JAPE 代码示例。

```
Rule:Coord
({Token.string == "and"} | {Token.string == "or"})
:andor
-->
:andor.Coordinate = {}

Rule:CoordAmbiguity
(
//{Split}
({Token.category == JJ})
({Token.category == NN} | {Token.category == NNS})
({Token.string == "and"} | {Token.string == "or"})
({Token.category == NN} | {Token.category == NNS})
//{Split}
):coord_ambiguity
-->
:coord_ambiguity.Coordinate = {}
```

表 1 给出了我们提出的用于标注自然语言需求缺陷类型的模式示例。在此基础上,我们通过人工的复查与修正,由前述的 4 名硕士研究生、博士研究生按照

缺陷类型定义分组对每种缺陷检测的结果进行是否符合缺陷定义的交叉评估。在对每一句需求语句检查确认了其缺陷标注情况以后,便可以将需求文档转化为如式(3)、式(4)定义的形式。

表 1　需求缺陷标签匹配模式

缺陷类型	模式
指代模糊性	P_{ANA} = (NP)(NP) + (Split)[0,1](Token. POS == PP|Token. POS = ~PR *)
并列模糊性	P_{CA} = (Token. POS == JJ)(Token. POS == NN|NNS)(Token. string == AND|OR)(Token. POS == NN|NNS)
模糊副词与短语	P_{AAP} = (Token. POS == RB|RBR),(Token. string = ~"[.] * ly $ ")
连词导致的非原子性	P_{NCO} = (Sentence) + (Token. kind == punctuation) * (Token. string == AND) + (Sentence)
从句导致的非原子性	P_{NCL} = (Sentence. len > 60)((Token. POS == CONJ) * (Token. string ∈ ClauseConn))
被动语态	P_{PV} = (AUXVERB)(NOT)?(Token. POS == RB|RBR)?(Token. POS == VBN)
条件缺失	P_{MC} = (IF)(Token,! Token. kind == punctuation) * (Token. kind == punctuation)(!(ELSE|OTHERWISE))

3.3　评估结果

最终能够完成缺陷类型标注的需求语句共计 1 101 句,我们根据式(5)-式(11)对标注后的数据集进行了评估计算。

缺陷覆盖指标上的评估结果统计如表 2 所示。可以看出,前文定义的缺陷分类得以被完全覆盖,也就是说,该数据集中的需求语句能够满足目前已有工作中对需求缺陷类型标签的需要。

表 2　需求缺陷类型覆盖情况

缺陷类型	缺陷覆盖率/%	覆盖标准差	覆盖偏差值
指代模糊性	17.89	0.110 3	50.52
并列模糊性	38.60		69.30
模糊副词与短语	20.16		52.58
连词导致的非原子性	9.90		43.27
从句导致的非原子性	6.09		39.81
被动语态	20.16		52.58
条件缺失	8.45		41.95

在 7 类需求缺陷中,最常见的缺陷类型是并列模糊性,出现率达 38.60%,而最少见的缺陷类型是从句导致的非原子性,仅占 6.09%。

从覆盖偏差值也可以看出,实际上缺陷类型的分布并不那么均衡,这些特征也与 PURE 数据集是从现实需求数据构建而来分不开关系,在现实需求文档中非原子性缺陷的出现率本身偏低,但并非完全不会出现。

缺陷类型之间的相对重叠度与数据集整体的全局重叠度如表 3 所示,可以看出在 PURE 数据集中两类重叠度都并不高,重叠度也与缺陷类型的覆盖率呈正相关。

表 3　需求缺陷类型重叠情况

相对重叠度	指代模糊性	并列模糊性	模糊副词与短语	连词导致的非原子性	从句导致的非原子性	被动语态	条件缺失
指代模糊性	—	0.090 8	0.051 8	0.031 8	0.030 0	0.041 8	0.023 6
并列模糊性	0.090 8	—	0.094 5	0.094 5	0.046 3	0.090 8	0.028 2
模糊副词与短语	0.051 8	0.094 5	—	0.030 0	0.017 3	0.040 0	0.020 0
连词导致的非原子性	0.031 8	0.094 5	0.030 0	—	0.040 0	0.023 6	0.002 7
从句导致的非原子性	0.030 0	0.046 3	0.017 3	0.040 0	—	0.018 2	0.005 4
被动语态	0.041 8	0.090 8	0.040 0	0.023 6	0.018 2	—	0.034 5
条件缺失	0.023 6	0.028 2	0.020 0	0.002 7	0.005 4	0.034 5	—
全局重叠度	0.040 7						

多缺陷比重的计算结果如表 4、图 2 所示,可以看出在标注后的 PURE 数据集中,依然是以无缺陷与只具有一种缺陷类型的需求语句为主,在缺陷的复合程度上属于较低。

表 4 需求缺陷类型的多缺陷比重情况

多缺陷数量	多缺陷比重	语句数量
0	0.341 5	376
1	0.317 0	349
2	0.196 2	216
3	0.096 3	106
4	0.034 5	38
5	0.010 9	12
6	0.003 7	4
7	0	0

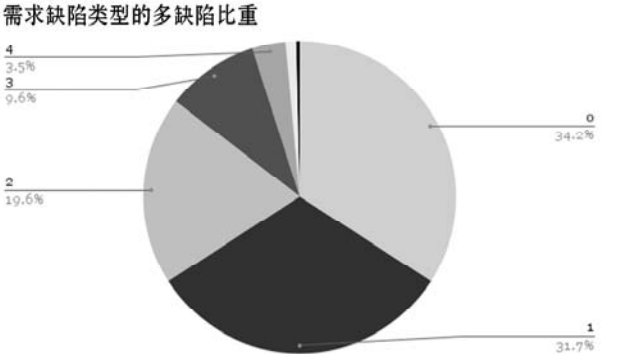

图 2 需求缺陷类型的多缺陷比重对比

总体而言,PURE 数据集基本可以满足需求缺陷检测任务的模型的需求,在缺陷标签上实现了完全覆盖,但在分布的均衡性上有所欠缺,这一点也值得未来的需求数据集构建工作注意。PURE 数据集上的缺陷类型的重叠度与复合程度都较低,这有助于它支撑模型的特征提取,但不适合训练更复杂的模型。此外,公开的需求文档还存在需求质量良莠不齐的问题,虽然 PURE 数据集的原始文档包含 79 篇需求文档,但其中相当一部分都无法通过预处理步骤并被正确地标注,可见提升需求文档本身的质量对于构建公开可用的数据集也是一个重要的方向。

4 结 语

需求通常由自然语言写成,并被认为是开发人员和客户之间进行交流的辅助工具。作为交流方式中最灵活的类型之一,自然语言需求也继承了自然语言本身天然的缺陷。这些缺陷可能导致系统开发过程中的问题。研究者们在已有的相关工作中提出了许多方法来帮助开发人员从质量的不同角度改进自然语言需求,但针对这一领域研究的需求缺陷定义的工作仍然没有达成共识,既缺少合适的、足够多的需求缺陷数据集,也缺少对需求缺陷数据集的评估框架。

本文综合考虑了过往有关需求质量的工作中涉及的需求缺陷,针对需求缺陷检测这一项任务,对现有的公开需求数据集能否满足此项任务进行了评估。本文首次提出了针对需求缺陷数据集的度量模型与评估框架,给出了需求数据的形式化定义与量化评估指标,同时设计了基于规则的框架,对现有需求数据集在需求缺陷任务方面的表现进行了量化指标计算与评估统计。结果表明,现有的公开数据集基本可以满足需求缺陷检测任务的模型的需求,在缺陷标签上实现了完全覆盖,但在分布的均衡性上有所不足,并在训练复杂模型的可能性上存在不足,而公开需求文档的质量问题也导致了可标注数据集在规模上的欠缺。

因此对于未来的工作,我们计划收集更多的公开需求文档为进一步的需求质量任务做准备,对现有的公开需求数据集进行扩充,并计划对需求质量任务中的不同算法模型进行对比,在算法上进一步进行改进。我们也计划与对自然语言处理感兴趣的相关研究人员和工业界专家合作以获取更多特定领域的需求数据与缺陷标准。

参 考 文 献

[1] Ferrari A, Spagnolo G O, Gnesi S. PURE: A dataset of public requirements documents[C]//25th International Requirements Engineering Conference,2017:502 - 505.

[2] Alhoshan W, Batista-Navarro R, Zhao L P. Towards a corpus of requirements documents enriched with semantic frame annotations[C]//26th International Requirements Engineering Conference,2018:428 - 431.

[3] McCoy J R. NASA software tools for high-quality requirements engineering[C]//26th Annual NASA Goddard Software Engineering Workshop,2001:69.

[4] Carlson N, Laplante P. The NASA automated requirements measurement tool: A reconstruction[J]. Innovations in Systems and Software Engineering,2014,10(2):77 - 91.

[5] Arora C, Sabetzadeh M, Briand L, et al. RUBRIC: A flexible tool for automated checking of conformance to requirement boilerplates[C]//9th Joint Meeting on Foundations of Software Engineering,2013:599 - 602.

[6] Huertas C, Juárez-Ramírez R. NLARE, A natural language processing tool for automatic requirements evaluation[C]//CUBE International Information Technology Conference,2012:371 - 378.

[7] Ambriola V, Gervasi V. Processing natural language requirements[C]//12th IEEE International Conference Automated Software Engineering,1997:36 - 45.

[8] Rosadini B, Ferrari A, Gori G, et al. Using NLP to detect requirements defects: An industrial experience in the railway domain[C]//International Working Conference on Requirements Engineering: Foundation for Software Quality,2017:

344 – 360.
[9] Gnesi S, Lami G, Trentanni G. An automatic tool for the analysis of natural language requirements[J]. Computer Systems Science and Engineering,2004,20(1):53 – 62.
[10] Ferrari A, Dell'Orletta F, Spagnolo G O, et al. Measuring and improving the completeness of natural language requirements[C]//20th International Working Conference on Requirements Engineering: Foundation for Software Quality, 2014:23 – 38.
[11] Arora C, Sabetzadeh M, Briand L, et al. Automated checking of conformance to requirements templates using natural language processing[J]. IEEE Transactions on Software Engineering,2015,41(10):944 – 968.
[12] Yang H, Roeck A, Gervasi V, et al. Analysing anaphoric ambiguity in natural language requirements[J]. Requirements engineering,2011,16(3):163 – 189.
[13] Saltzer J H, Kaashoek M F. Principles of computer system design: An introduction[M]. Amsterdam: Morgan Kaufmann,2009.
[14] Lauesen S. Software requirements: Styles and techniques [M]. New York: Pearson Education,2002.
[15] Toutanova K, Klein D, Manning C D, et al. Feature-rich part-of-speech tagging with a cyclic dependency network [C]//Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics,2003:252 – 259.
[16] Marcus M, Santorini B, Marcinkiewicz M A. Building a large annotated corpus of English: The Penn treebank[M]. Cambridge: MIT Press,1993.
[17] Open NLP. Apache software foundation[EB/OL].[2022 – 06 – 14]. http://opennlp.apache.org.
[18] Finkel J R, Grenager T, Manning C D. Incorporating non-local information into information extraction systems by Gibbs sampling[C]//43rd Annual Meeting of the Association for Computational Linguistics,2005:363 – 370.
[19] Sang E F, Meulder F. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition [EB]. arXiv:0306050,2003.
[20] Chinchor N A. Overview of muc-7/met-2[EB/OL].[2022 – 06 – 14]. https://aclanthology.org/M98-1001.pdf.
[21] Sharafaldin I, Lashkari A H, Ghorbani A. Toward generating a new intrusion detection dataset and intrusion traffic characterization[C]//4th International Conference on Information Systems Security and Privacy,2018:108 – 116.
[22] Oh S. A new dataset evaluation method based on category overlap[J]. Computers in Biology and Medicine,2011,41 (2):115 – 122.
[23] Röder M, Usbeck R, Hellmann S, et al. N3-A collection of datasets for named entity recognition and disambiguation in the NLP interchange format[C]//International Conference on Language Resources and Evaluation,2014:3529 – 3533.
[24] Sharghi A, Laurel J S, Gong B. Query-focused video summarization: Dataset, evaluation, and a memory network based approach[C]//IEEE Conference on Computer Vision and Pattern Recognition,2017:4788 – 4797.

(上接第 72 页)

来了新一轮发展曙光与契机,打造自主可控、众创共享的复杂数值模拟系统的需求日益增加。尽管目前存在着核心软件受制于人、高度依赖专家经验、模拟仿真技术应用分散、信息/数据流不连续等问题,本文提出通过应用模式和开发模式的创新,建立松耦合的系统架构和"众创共享"的开发模式,实现国内船舶行业优势力量的工业知识(App)和数据的汇聚,打造面向智能化应用的总体性能数值模拟服务系统的方案,为逐步解决我国船舶工业中设计仿真类软件存在的卡脖子问题,进而有力支撑我国船舶行业的转型,作出了有益探索。

参 考 文 献

[1] 赵峰,吴乘胜,黄少锋,等. 数值水池路线图[J]. 船舶力学,2014,18(8):924 – 932.
[2] Hurwitz M. The CREATE program/CREATE-SHIPS project: Examining the S&T enterprise in naval engineering national academy of sciences[EB/OL].[2021 – 05 – 18]. https://ndiastorage.blob.core.usgovcloudapi.net/ndia/2011/physics/TuesdayHURWITZ.pdf.
[3] Gorski J. CREATE-Ships Hydrodynamic Products [EB/OL].[2021 – 05 – 18]. https://ndiastorage.blob.core.usgovcloudapi.net/ndia/2011/physics/TuesdayHURWITZ.pdf.
[4] 赵峰,吴乘胜,张志荣,等. 实现数值水池的关键技术初步分析[J]. 船舶力学,2015,19(10):1209 – 1220.
[5] 李琴. 中国船舶工业的软件之忧[EB/OL].[2021 – 05 – 18]. https://mp.weixin.qq.com/s/n0nryYDBHhFeYC0DsDaDhw.
[6] 朱焕亮,徐保温. 工业软件浅析[J]. 航空制造技术,2014 (18):22 – 27.
[7] 徐恒,吴丽琳. 工业软件企业发展及对策建议—基于研发设计类软件开发[J]. 工业技术创新,2019,6(1):99 – 106.
[8] 陈琛. 构建先进制造的智慧海洋—在科学认识工业软件产业发展规律的基础上谈国产工业软件的突破之路[EB/OL].[2021 – 05 – 18]. https://card.weibo.com/article/m/show/id/2309404497748874362900.
[9] 程庆和,伍英杰. 面向智能制造的船舶行业自主工业软件发展探讨[C]//数字化造船学术交流会议,2018.
[10] 赵峰,陈伟政,韦喜忠,等. 系统工程在船舶总体性能研发中的实践思考[J]. 中国造船,2021,62(2):275 – 283.
[11] 韦喜忠,金建海,王墨伟,等. 面向船舶总体性能预报 APP 研制的 GJB5000A 应用方案[J]. 船舶标准化工程师,2020,53(4):5 – 10.