6L-07

組込みシステムに関連する README ファイルの必要情報の調査と不足情報警告ツールの開発

鳥木　瑛司　　山本　椋太　　阿部　司

苫小牧工業高等専門学校

１．はじめに

近年，企業でオープンソースソフトウェア（以下，OSS）を利用するケースが増えている．

OSS を利用する際，ユーザは README ファイル（以下，README）を読み，そのリポジトリに関する情報を取得する．GitHub Docs は，README が含むべき内容を，プロジェクトが行うこと，プロジェクトの有益さ，プロジェクトの始め方，サポート体制，貢献・維持の情報としている [1].

一方，README に関する問題として，情報が不足している README が多いという現状がある[2] .

本研究では，README の不足情報を警告するツールを開発し，評価を行う．ツールの開発前に，ツールとして警告するべき不足情報を手作業で調査する．なお，本研究ではハードウェアのような特有の情報を含む，組込みシステムの README を対象とする．

２．必要記述項目の調査

調査は，README の必要記述項目を明らかにすることを目的とし，以下の手順で実施する．

1. GitHub 上で Arduino を検索語としたときの検索結果である README を 50 件取得する．
2. 第 1 著者が README の目視調査を行う．調査では，GitHub Docs が推奨する記述項目の有無を判定する[1].
3. 第 2・3 著者との議論で，README の必要記述項目を定める．

この調査より，必要記述項目は以下の 2 点である．

項目 1：開発者が実現したこと

項目 2：ユーザが使用するための情報

項目 1 は他の開発成果物との差分を明確にするために，項目 2 はユーザが自身の環境で開発成果物を再現するために必要である．

提案ツールでは，項目 1 に対応する What, 項目 2 を詳細化した Hardware, Wiring, Software, Usage, Reference, License の計 7 つを必要記述項目とする．

The Investigation of Necessary Description for Embedded System README FIles and The Development of a Detection Tool for a Lack of the Description

Eiji Toriki, Ryota Yamamoto, Tsukasa Abe,
National Institute of Technology(KOSEN), Tomakomai College

３．README の不足情報警告手法

３．１　README の不足情報警告手法の実装

本手法は，①TF-IDF 辞書構成部と②欠陥抽出部に大別される．ツールの全体像を**図 1** に示す．

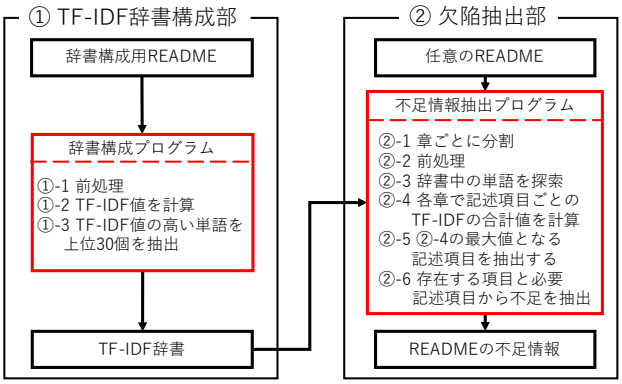

図 1　ツールの全体像

①の入力には，辞書構成用 README として，GitHub 上で Arduino と検索した際の検索結果から得た調査とは別の 46 件の README を用いる．これらを章ごとに分割し，記述項目ごとに分類したデータを入力する．なお，章とは Markdown の ‘#’, ‘---’, ‘===’ で始まる見出しから次の見出しの直前またはファイルの終端までの記述を指す．

①の出力は TF-IDF 辞書である．この辞書には，特定の文章で頻繁に出現する単語に与える手法である TF-IDF 法[3] によって，得られる重みの上位 30 単語を，記述項目ごとに格納する．TF-IDF 法による重み w_{td} は，以下の式によって得られる．

$$w_{td} = tf(t,d) \times idf(t) \qquad (1)$$

$tf(t,d)$は，文章 d 中における単語 t の出現頻度であり，$idf(t)$ は出現する文章数が少ない単語 t ほど大きな重みを与える．

②の入力は，不足を警告する README である．出力は，入力 README の不足項目である．不足項目は，全ての必要記述項目と，入力 README で存在する記述項目との差である．**図 1** より，提案手法では，入力 README を章ごとの記述に分割し，章ごとに必要記述項目を 1 つ割り当てる．必要記述項目の割り当ては，その章における TF-IDF 法の重みの合計値を用いる．重みが最大である記述項目をそ

表１　提案ツールの評価値

	ツール全体			What			Hardware			Wiring		
	P	Re	F	P	Re	F	P	Re	F	P	Re	F
A	0.817	0.872	0.843	0.143	0.333	0.200	0.920	0.821	0.868	0.958	0.742	0.836
R	0.695	0.884	0.778	0.143	0.667	0.235	0.808	0.808	0.808	0.941	0.914	0.928
M	0.618	0.790	0.694	0.200	0.500	0.286	0.529	0.692	0.600	0.903	0.824	0.862

	Software			Usage			Reference			License		
	P	Re	F	P	Re	F	P	Re	F	P	Re	F
A	0.923	0.923	0.923	0.939	0.969	0.954	0.600	0.913	0.724	0.941	0.970	0.955
R	0.846	0.786	0.815	0.417	1.000	0.588	0.531	0.810	0.642	1.000	0.970	0.985
M	0.769	0.556	0.645	0.300	0.818	0.439	0.444	0.800	0.571	1.000	1.000	1.000

※　P: Precision, Re: Recall, F: F 値, A: Arduino, R: Raspberry Pi, M: Mbed を表している．

の章に割り当て，存在する項目を抽出する．

なお，①②に共通する前処理では，形態素解析，ステミング，stop words の除去を行い，いずれも nltk[4] を用いて実装する．

３．２　README の不足情報警告手法の評価

欠陥抽出手法の評価は，ツールによる入力 README の不足と，筆者が割り振った入力 README の不足を比較することにより行う．入力 README は，GitHub 上で検索語を Arduino, Raspberry Pi, Mbed とした際の検索結果から得た README 各 40 件である．各 40 件のファイルサイズの合計は，Arudino が 104.4kB, Raspberry Pi が 180.0kB，Mbed が 139.3kB である．評価値は以下の 3 つである．

$$Precision = \frac{\text{ツール・正解が欠陥とした記述項目数}}{\text{ツールが欠陥とした記述項目数}} \quad (2)$$

$$Recall = \frac{\text{ツール・正解が欠陥とした記述項目数}}{\text{正解が欠陥とした記述項目数}} \quad (3)$$

$$F_{score} = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (4)$$

なお，提案ツールでは，Precision よりも Recall を重要視する．Recall が低いことは，欠陥を見逃していることに繋がるため，情報不足を警告するツールの目的に反するからである．(2)から(4)式で求めた，ツールの評価値を**表１**に示す．評価値算出のための記述項目数は紙面の都合上省略する．

４．考察

表１より，すべてのプラットフォームにおける Recall が 8 割程度の値となっている．一方で，Raspberry Pi・Mbed の Precision や F 値は，Arduino と比較して 1 割から 2 割程度低い．

また，**表１**より記述項目ごとの評価値から，What と Usage に問題があると分かった．

What において，3 つのプラットフォームの評価値はツール全体の評価値と比較して 5 割以上低い．What には多くの場合，固有名詞が書かれる．しかし，本手法において固有名詞は考慮されないため，What の特徴を正しく抽出できないと考える．

Usage は，Arduino 以外の 2 つのプラットフォームで Precision が 5 割を超えず，F 値も 5 割前後とツール全体と比較して 2 割から 3 割程度低い．原因は，Usage に記載される情報の違うからであると考える．Arduino では 87.5%の Usage にサンプルプログラムが記載されていたが，Raspberry Pi や Mbed では 10%から 20%程度であった．

よって，提案ツールは README 中の不足を正しく警告する割合が全体で 8 割程度である．一方，記述項目 What, Usage がツール全体の評価を低減する原因となっていると考える．また，妥当性への脅威として，本ツールは，別の組込みシステムに関するリポジトリの README を適用した際の評価は不明であることや，README 以外のファイルの欠陥は抽出できないことが挙げられる．

５．おわりに

本稿では，README の必要記述項目の調査と不足情報警告ツールの評価について述べた．

調査の結果として，README には，開発者が実現したことと，使用方法の記載が重要であることが分かった．欠陥抽出ツールは REAME 中の不足を正しく警告できる割合が 8 割であること，記述項目 What，Usage に問題があることが分かった．

今後は，提案ツールの評価の妥当性を高めるために，関連研究の調査，および比較を行う．

参考文献

[1] GitHub.com – GitHub Docs, https://docs.github.com/en/github
[2] Open Source Survey, https://opensourcesurvey.org/2017/
[3] 奥村 学: 自然言語処理の基礎 コロナ社, pp116-119，2010.
[4] Bird S., Loper E.：NLTK: The Natural Language Toolkit,In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics.2004