Keywords-based test categorization for Extra-Functional Properties

Accepted for publication in ICSTW: 4th International Workshop on Testing Extra-Functional Properties and Quality Characteristics of Software Systems (ITEQS)

Muhammad Abbas[1,2], Abdul Rauf[1], Mehrdad Saadatmand[1,2], Eduard Paul Enoiu[2], and Daniel Sundmark[2]
[1]RISE Research Institutes of Sweden, Västerås, Sweden
[2]Mälardalen University, Västerås, Sweden
[1]firstname.lastname @ ri.se ,[2]firstname.lastname @ mdh.se

***Abstract*—Categorizing existing test specifications can provide insights on coverage of the test suite to extra-functional properties. Manual approaches for test categorization can be time-consuming and prone to error. In this short paper, we propose a semi-automated approach for semantic keywords-based textual test categorization for extra-functional properties. The approach is the first step towards coverage-based test case selection based on extra-functional properties. We report a preliminary evaluation of industrial data for test categorization for safety aspects. Results show that keyword-based approaches can be used to categorize tests for extra-functional properties and can be improved by considering contextual information of keywords.**

Keywords–*test categorization, topic model, keyword extraction*

I. Introduction

Extra-Functional Properties (EFPs) of the system define the physiognomies of the system and can be crucial to the system's success [1]. These EFPs are realized by implementing functional requirements satisfying some "extra-functional" constraints (e.g, Every main functionality should be reachable in no more than three clicks). System correctness can be achieved with achieving the required level of functional correctness conforming to certain constraints. Software testing is one possible solution to check the conformance of the system to the specifications and these so-called extra-functional constraints. Testing for extra-functional properties is an active area of research [2]. EFPs can't be treated independently and thus testing (ideally) should include them. Unfortunately, testing systems rigorously consumes a significant amount of resources making it impossible to execute all possible test cases, due to time and budget limitations [3].

The selection of a subset of test cases (for execution) might be required to limit the scope of the testing. The test case selection based on certain criteria (for example coverage to EFPs [4]) for the System Under Test (SUT) is a key research area [5], [6]. Test case selection can be optimized for more than one criterion. Recently, a wide variety of learning and optimization based techniques have been employed to select a subset of test cases. EFPs are not atomic and thus executing some of the functional tests might provide coverage to an EFP.

In practice (in some cases), the test cases are first written (in natural language) as test specifications, describing how to test a particular requirement. Each test specification is usually linked to a requirement but several other requirements might be indirectly dependent on the tests originating from the specification. Identifying test specifications that might provide coverage to an EFP can be done at the test specification-level and this categorization can guide the test classification, selection, and prioritization process to reduce testing efforts. Natural Language Processing (NLP) and topic modeling can be used to categorize such test specifications for EFPs automatically.

The use of NLP and topic modeling in testing is an active area of research [7]–[10]. A lot of research has been focused on test case generation from requirements written in (controlled) natural language [11]. Literature also reveals applications of such approaches in malicious application detection [8], test case generation, [9] and dependencies based test scheduling [12]. Test categorization and classification (at textual-level) can also benefit from NLP and topic modeling.

Existing approaches are using NLP and topic modeling for test specifications (tests from now on) prioritization and classification [13], [14]. Thomas et al. based their work [13] on the hypothesis that tests sharing common topics are most likely going to be functionality-wise similar. This hypothesis was used to guide test case prioritization based on maximum functionality coverage. Topic modeling is also used to coordinate requirements and testing activities [15]. Another study [14] aims at automating the feature labeling of test cases using topic modeling. The approach suggests possible feature labels (tags) for mobile application tests. Such keywords based tagging is also used in the test management tool TestLink [1]. To the best of our knowledge, no existing work investigates semantic keyword-based test case categorization for EFPs.

Contributions. In this paper, we reported an approach for keywords-based test categorization for extra-functional properties. We hypothesize that the EFPs might be associated with certain abstract topics and those topics can be extracted from already categorized documents (such as requirements, and standards, etc.) automatically. Each topic is a cluster of words and thus we extract relevant keywords from each extracted topic to derive keyword dictionaries. Rapid Automatic Keywords Extraction (RAKE) algorithm [16] is then used to derive keywords from tests and categorize them based on

[1]TestLink: http://testlink.org/

intersection score to EFPs' dictionaries. We also hypothesize that the terminological coverage of tests (to an EFP) is a predictor for actual coverage to the EFPs. As proof of concept, we applied our approach to already categorized tests from our industrial partner (Bombardier Transportation AB).

Structure. The rest of this paper is structured as: Section II discussed our proposed approach for tests categorization, Section III presents the evaluation and discusses the preliminary results, Section IV discusses relevant threats to validity of our results, and Section V concludes the paper with future directions.

II. Proposed Approach

Our test specification categorization approach (shown in Figure 1) has two steps. In the first step, our approach builds abstract keyword dictionaries for EFPs. In the second step, our approach extracts relevant keywords from the test specification and computes the closeness score of the test specification to the dictionaries of EFPs. In this section, we discuss the steps in detail.

a) Dictionaries Extractor: As discussed, in practice the EFPs are associated with different assets/components (keywords) and have a domain-specific interpretation. The associated keywords to an EFP can be extracted from already classified documents (such as non-functional requirements) using NLP techniques. Our approach uses the existing and already classified documents to extract EFPs' keyword dictionaries. This is done by taking n sets of labeled (EFP it belongs to) documents. The input documents are cleaned by removing all English stop words (such as shall, will, etc.). The cleaned documents are converted into lower case and each token of the document is tagged with a parts-of-speech (POS) tags. This step is necessary for lemmatization. The tagged tokens of each document are converted to their lemmas (base). Converting the words into their base words aids avoiding treatment of the same word differently (e.g, interface and interfacing both have the same base i.e: interface). After this step, term frequency based vectors are extracted from the cleaned documents. We use a Bag-of-Word (BoW) model to computes the frequency vectors for each document. The vectors are then used to fit a generative topic model to extract topics from the frequency vectors. In our case, we use Latent Dirichlet Allocation (LDA) to extract the top nine abstract topics from each set of documents. Each topic is a combination of keywords extracted from the documents. Top ten words from each topic are considered for the construction of the dictionary and n set of labeled (as per EFP) dictionaries are produced as an output. A manual review of the dictionaries is performed to remove irrelevant words and add relevant words that were extracted by LDA but were not top. The manual step is necessary to ensure the quality of the extracted dictionaries. Note that some tools heavily rely on user-defined keywords and manual tagging. We believe such tools can benefit from approaches like ours.

b) Test Categorizer: Our approach considers the fact that a test can provide coverage to more than one EFPs and thus calculates the intersection score (a value between 0 to 1) for each EFP's dictionary. The intersection score represents the closeness of the input document to a specific dictionary. The intersection score is calculated by extracting keywords from the cleaned input test specifications. The cleaning is performed by removing the stop words, POS tagging the tokens, and finally applying the RAKE algorithm. The extracted keywords are lemmatized and are passed to the intersection score calculator. The intersection score is calculated per dictionary by dividing the number of common words in the test specification over the total number of keywords ($\cap_{score} = K_{matched}/len(K)$, where K is the number of keywords in the test and $K_{matched}$ is the number of common keywords in a given dictionary and a given test specification.). The intersection score per test case is calculated against each dictionary and the labels for the dictionary with an intersection score greater than a threshold is selected as the possible class(es) (EFPs based categories) for the test case specification.

III. Evaluation

This section of the paper discusses the application of the keyword-based text categorization approach to data from a large company producing vehicular embedded systems. Some aspects and data provided by this company have been anonymized in the following sections.

A. Research Questions

We aim to answer the following research questions:

RQ1. *Is keyword-based test categorization applicable for categorizing tests for EFPs?* We answer this research question by applying our approach to categorize tests for safety aspects of the system.

RQ2. *What is the accuracy of the keyword-based test categorization selection for Safety EFP?* We answer this research question by reporting the accuracy of our approach on a small data-set of 20 tests.

B. Implementation

We implemented a prototype tool in Python. The dictionaries extraction part of the tool takes a spreadsheet as input with the text of each document per row. An LDA model (from Gensim [17]) is fitted to the pre-processed (using spaCy [2]) and vectorized documents with 10 $components$. From each topic, the top ten words or phrases are extracted and a dictionary is produced as an output. The `Test Categorizer` takes in the test specification as input in the spreadsheet and uses RAKE [3] for keyword extraction. The Rake algorithm was configured to extract keywords containing no more than two words. The extracted keywords from the test specifications are searched in each dictionary and the intersection score is calculated. Note that the current prototype does support comparison with multiple dictionaries however for this evaluation we only considered safety. A predefined threshold (on the intersection score) is required as an input to assign a test to a particular

[2]Industrial-Strength NLP, https://spacy.io/

[3]RAKE-NLTK https://pypi.org/project/rake-nltk/

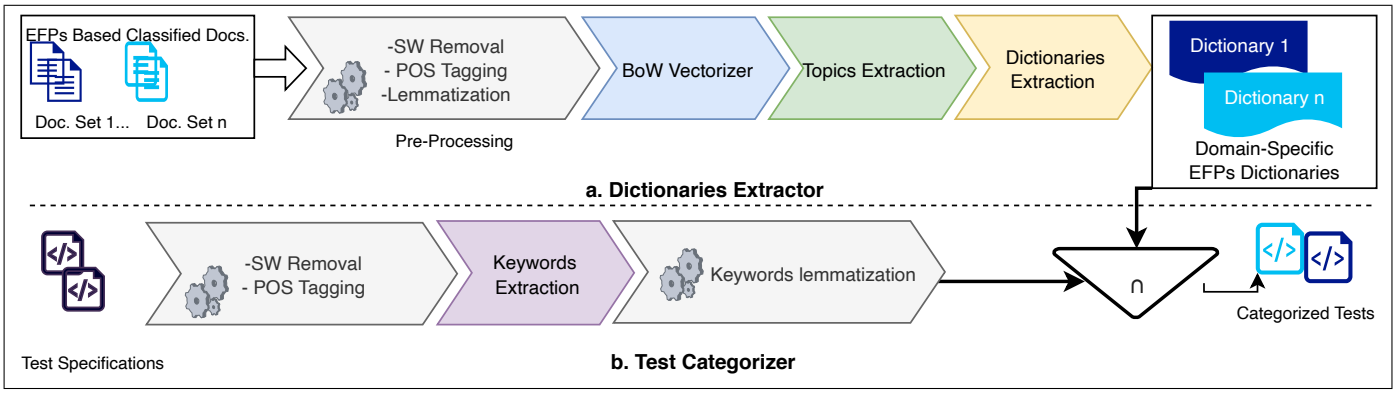

Fig. 1. Proposed Keyword-based Test Specifications Categorization Approach

category. In our case, we categorized a test as safety if the intersection score is greater than zero.

C. Data Preparation

For the first step (`Dictionaries Extractor`), we selected safety requirements from the domain of Propulsion Control in the railway industry. We selected all the safety-related requirements which were already classified to be safety-critical. We ended up with 28 high-level safety requirements. For effective topic modeling, we combined two requirements to create one input document (total 14) for LDA. For the second step (`Test Categorizer`), 20 test specifications were selected for this evaluation. Note that these tests were high-level specifications and in code-level, one test specification might be linked to tens of different test cases at different-levels. Fifteen out of the 20 test specifications were marked as safety-related by experts. Each considered test was a combination of Test Objective, Test Setup, Test Sequence, and Test Acceptance Criteria. All the text in different parts of the test is combined and is considered as one test specification. In total, we selected 28 tests as an input to the second step of our approach.

D. Metric for Evaluation

We used accuracy for evaluation of our approach. Accuracy, in our case, is treated as a ratio between the number of correct categorization and the total number of input test specifications to the `Test Categorizer`. A categorization is considered correct if our approach marks the test as safety-related and the test is marked as safety-related by the experts too (ground-truth).

E. Procedure

To demonstrate the applicability of our approach, we extracted a dictionary (for safety) from the safety requirements. The dictionary contained keywords and phrases extracted from the LDA topics. The dictionary was manually cleaned and some domain related safety keywords were added (e.g, overheat, under-voltage, etc.). This step was necessary since the dictionary was extracted from a sub-set of requirements that were available to us. Based on the input documents the interpretation of safety (as word cloud) is shown in Figure 2. After this step, we categorized the 20 test specifications and calculated the intersection score. If the intersection score was greater than zero, the test specification was considered to be a safety test. Table I shows the summary of the results of the test categorization.

TABLE I
SUMMARY OF THE TESTS CATEGORIZATION

Safety		Non-Safety		**Total**	
Correct	Incorrect	Correct	Incorrect	Correct	Incorrect
15	0	2	3	17	3

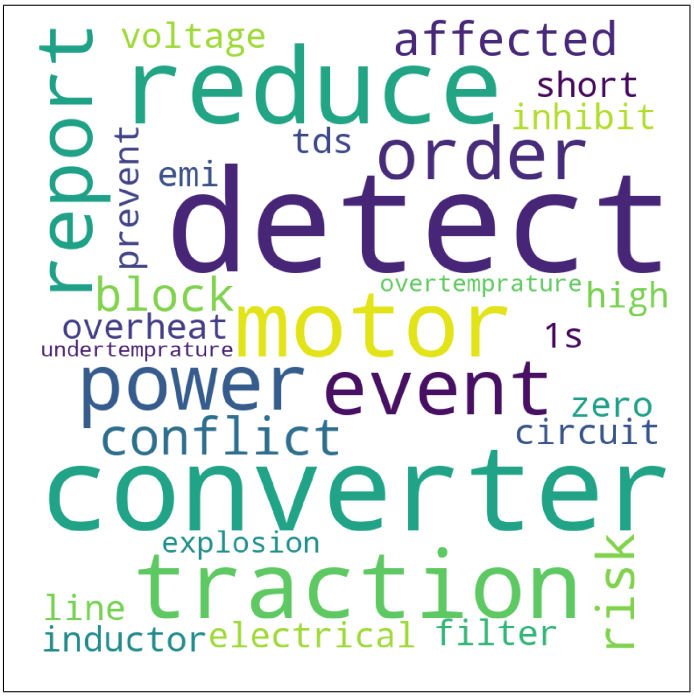

Fig. 2. Word Cloud of Safety Dictionary

F. Preliminary Results & Discussion

To answer **RQ1**, we applied our test categorization approach to industrial test specifications. We demonstrated the applica-

bility of our keywords-based approach for test specifications. Results show that keyword-based approach can effectively categorize test specifications written in natural language. However, the results of keyword-based categorization can be heavily dependent on the relevant keywords in the dictionaries. **RQ2.** We found that for safety, our approach categorized the given test specifications with 100% accuracy. This is because at-least one of the keywords (extracted from the test) for each test is found in the safety dictionary. However, the same keywords were also found in three non-safety test specifications. It is because our approach does not consider the context in which a keyword is used. In one incorrect non-safety case we found that the test specification is using a safety-related function in the test sequence but the test was not marked as a safety test. Meaning that the test specification is not directly providing any coverage to the safety function but is somehow dependent on it. We believe that such approaches can be beneficial in identifying such cases.

IV. Threats to Validity

Here we address some relevant threats to the validity of our results. We treated the problem of test categorization as a keywords based filtering problem. Such approaches might result in a high false-positive rate. To tackle this *construct validity* threat, we introduced an intersection score. An intersection score represents the percentage of common keywords with a dictionary and can be used to reduce the false positives. However, the quality of the keywords in the dictionaries can significantly effect the end results of our approach. We recommend a manual review of the generated dictionaries to overcome this problem.

We used a very small data-set intending to categorize the tests for just one extra-functional property (safety). Tackling this *external validity* threat needs further investigations on relatively larger data-sets to categorize tests for multiple EFPs.

V. Conclusion & Future Work

In this short paper, we proposed a keywords-based test categorization approach. To demonstrate the applicability and present the preliminary results, we applied our approach to industrial test specifications to categorize safety tests. Our results show that our approach can be used to categorize test specifications per extra-functional property. We found that such an approach can also be useful to identify test cases that are indirectly providing coverage to an EFP. However, we also noted that keywords-based test categorization can lead to a high false-positive rate since the dictionaries' keywords can be found in tests not related to EFPs. The false-positive rate can be tackled by choosing a suitable threshold for the intersection score. Further investigation (on larger data-set) is required to report on the effectiveness of the approach.

Our future work includes extending this approach to also consider the context of the used keywords to reduce the false-positive rate. Variability-aware test case prioritization and selection based on keywords is also one of our future focus. Lastly, we are also investigating the use of common keywords-based dependencies detection in natural language requirements and test specifications.

Acknowledgment

This work has been supported by and received funding from the XIVT and ARRAY Projects. The authors would also like to thank people at Bombardier Transportation AB, for their continued support.

References

[1] J. Cleland-Huang, "Quality requirements and their role in successful products," in *Proceedings - 15th IEEE International Requirements Engineering Conference, RE 2007*, 2007, p. 361.

[2] W. Afzal, R. Torkar, and R. Feldt, "A systematic review of search-based testing for non-functional system properties," *Information and Software Technology*, vol. 51, no. 6, pp. 957–976, 2009.

[3] G. J. Myers, C. Sandler, and T. Badgett, *The Art of Software Testing*, 3rd ed. Wiley Publishing, 2011.

[4] M. Abbas, I. Inayat, M. Saadatmand, and N. Jan, "Requirements Dependencies-Based Test Case Prioritization for Extra-Functional Properties," in *2019 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*. Xi'an, China: IEEE, Apr. 2019, pp. 159–163.

[5] R. Kazmi, D. N. A. Jawawi, R. Mohamad, and I. Ghani, "Effective regression test case selection: A systematic literature review," *ACM Comput. Surv.*, vol. 50, no. 2, May 2017.

[6] S. Yoo and M. Harman, "Regression testing minimization, selection and prioritization: a survey," vol. 22, no. 2, pp. 67–120.

[7] C. Wang, F. Pastore, A. Goknil, and L. C. Briand, "Automatic generation of system test cases from use case specifications: an nlp-based approach," 2019.

[8] A. Gorla, I. Tavecchia, F. Gross, and A. Zeller, "Checking app behavior against app descriptions," in *Proceedings - International Conference on Software Engineering*, no. 1, 2014, pp. 1025–1035.

[9] G. Carvalho, D. Falcão, F. Barros, A. Sampaio, A. Mota, L. Motta, and M. Blackburn, "NAT2test SCR : Test case generation from natural language requirements based on SCR specifications," *Science of Computer Programming*, vol. 95, pp. 275–297, Dec. 2014.

[10] S. Masuda, T. Matsuodani, and K. Tsuda, "Automatic Generation of UTP Models from Requirements in Natural Language," *Proceedings - 2016 IEEE International Conference on Software Testing, Verification and Validation Workshops, ICSTW 2016*, pp. 1–6, 2016.

[11] I. Ahsan, W. H. Butt, M. A. Ahmed, and M. W. Anwar, "A comprehensive investigation of natural language processing techniques and tools to generate automated test cases," in *Proceedings of the Second International Conference on Internet of Things, Data and Cloud Computing*, ser. ICC '17. ACM, 2017.

[12] S. Tahvili, L. Hatvani, M. Felderer, W. Afzal, M. Saadatmand, and M. Bohlin, "Cluster-based test scheduling strategies using semantic relationships between test specifications," in *Proceedings of the 5th International Workshop on Requirements Engineering and Testing*. ACM, 2018, pp. 1–4.

[13] S. W. Thomas, H. Hemmati, A. E. Hassan, and D. Blostein, "Static test case prioritization using topic models," *Empirical Software Engineering*, vol. 19, no. 1, pp. 182–212, Feb. 2014.

[14] J. Shimagaki, Y. Kamei, N. Ubayashi, and A. Hindle, "Automatic topic classification of test cases using text mining at an Android smartphone vendor," in *Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement - ESEM '18*. Oulu, Finland: ACM Press, 2018, pp. 1–10.

[15] M. Unterkalmsteiner, "Coordinating requirements engineering and software testing," Ph.D. dissertation, 06 2015.

[16] S. Rose, D. Engel, N. Cramer, and W. Cowley, *Automatic Keyword Extraction from Individual Documents*. John Wiley & Sons, Ltd, 2010, ch. 1, pp. 1–20.

[17] R. Řehůřek and P. Sojka, "Software Framework for Topic Modelling with Large Corpora," in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Valletta, Malta: ELRA, May 2010, pp. 45–50.