

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/341655455>

Null function classification using NFR matrix and decision tree

Research · January 2018

DOI: 10.13140/RG.2.2.25854.82246

CITATIONS

0

READS

151

2 authors:



[Hedayat Hoshmand](#)

Chandigarh University

2 PUBLICATIONS 1 CITATION

[SEE PROFILE](#)



[Sandeep Singh Kang](#)

Chandigarh University

119 PUBLICATIONS 1,348 CITATIONS

[SEE PROFILE](#)

Null function classification using NFR matrix and decision tree

Hedayatullah Hoshmand*, Sandeep Singh Kang

Department of Computer science and engineering, Chandigarh University, Mohali, Punjab, India

*Corresponding author E-mail: hedayatullah.hoshmand@gmail.com

Abstract

This research work is based on the classification of the functions which are required and which is not required for the efficient development of the software. The motivation of this work is to identify non required functions to reduce development cost and efforts. The classification is the technique which can classify data into certain number of classes. The NFR matrix is the existing technique which can classify the Null functions. To classify the null functions NFR use clustering method which can modify to increase accuracy of classification. In the existing method NFR matrix is used for the null function classification, it use the clustering for the classification. In this work, decision tree will be based with the clustering. It takes input result of clustering and generates classified result.

Keywords: Classification; Null Function; Decision tree; NFR matrix

1. Introduction

An engineering application that is applied in order to develop software by following a systematic mechanism is known as software engineering. The development as well as maintenance of software is ensured to be running in an efficient manner through this approach. Within several home applications, automobiles as well as airplanes, software is involved. Also, there will always be an increase in demand of software within further applications as well due to the consistent increase in blurriness of boundaries amongst telecommunications, entertainment as well as the computer industries [1]. If an organization needs to add a complex and high quality of software at affordable cost, the usage of software engineering is must. In order to provide software development process, the more subtle influence of Internet is also important. Teams are formed which help in software development due to the increase in complexity of modern software systems. Without having a basic understanding about what is the real purpose of software system that is being developed, the detailed designing as well as testing cannot be performed [2]. Also, the requirements that are mainly the services performed by the software system need to be specified at earlier stages. The services that are to be provided by software and the constraints within which it needs to operate are known as requirements. There can either be high-level abstract statements of services present within the requirements or some mathematical functional specifications that have detailed system constraints within them [3]. The services that are required by the client are established along with the constraints required to develop and operate the systems are provided through requirements engineering process [4]. The requirements are broadly classified as functional and non-functional. Through the description of processing of information or materials as inputs or outputs, the requirements of users are directly supported within the functional requirements. The specification regarding what the software must do

is defined by functional requirements [5]. The actions that need to be performed such that the fundamental reasons for existence of software can be satisfied are mainly provided here. These types of requirements are much similar to that of the business requirements [6]. The description of things that need to be completed in order to perform certain actions can be described by the user involved. Thus, the complete description of actions or steps that need to be followed in order to design software is known as functional requirement of software. The description of business standards and business environment along with the overall experience of a user such that all the users are supported are provided through non-functional requirements [7]. The characteristics that must be included within the software are thus defined through NFR. For example- the characteristics or qualities through which the product can be made more attractive, usable, faster or reliable are required within a product. These activities are to be ensured within the product to ensure that the fundamental activities of the product can be performed [8]. They ensure that a software system works in the required manner only but are not the part of fundamental reason for the existence of that system completely. The functionality of software is not interrupted with the involvement of NFRs [9]. Irrespective of the properties that are attached to the functional requirements, there are no changes made in them. Functionality is added to the software with the addition of NFRs. In order to ensure that the software is easy to use, secure as well as interactive, there are some efforts needed to be made. Thus, the requirements that perform actions are known as functional requirements whereas those that provide characters to the software are known as non-functional requirements [10]. Within the specification, an important part involved is provided by the NFRs. Since the product is judged completely on the basis of NFRs, they play an important role within the client as well as user.

There has been an increase in demand of automatically classifying the documents into predefined categories. The broader classification methods in which the data can be classified are as unsupervised, supervised and semi-supervised methods [11]. There has been extensive research done on automatic text classification task lately. Several machine learning approaches have also been involved here in order to provide enhancements [12]. There are numerous classifiers proposed that are used during the classification process. A learning algorithm that uses distance or similarity functions for attaining pair of observations is known as K-NN classifier [13]. Due to the effectiveness, non-parametric as well as ease of implementation, many applications have utilized this method. The longer duration of classification time as well as calculation of optimal value of k are however its disadvantages. Another classifier that is known under priori probability as well as the class conditional probability is the Naïve bias classifier [14]. The probability that a document D belongs to class C is computed through this classifier. Multivariate Bernoulli and multinomial model are the two event models presented for Naïve bias. In case when the database is huge, the model multinomial model is utilized. However, rough parameter estimation and the presence of only few training documents are the two major issues that arise through this method [15]. Decision tree classifier is another type of classifier used for classification. Here, the label is presented through the internal node, the test on weight is labeled through the branches departing from the internal node and the relevant class labels are represented in the leaf node. The running is performed in order to classify the document which initialized from the root and then reaches to specific leaf. The objective of classification of document is then represented here. In order to classify the text, another application used is Support Vector Machine (SVM) [16, 17]. There are both positive as well as negative training sets provided within SVM which are not found in order methods. In order to provide a decision surface through which the positives can be differentiated from negative data within the n -dimensional space otherwise known as hyper plane, these two positive and negative training sets are used.

2. Related Work

Fatima Khalique, et al. [18] proposed a mechanism for generation of Domain Non-Functional Requirements (DNFRs) by utilizing model driven approach. In order to automate the generation of DNFR, two model transformations are performed. A PNFR meta-model is utilized in order to convert PNFR into Product Line Non-Functional Requirements (PLNFR) within the initial transformation. Further, DNFR meta-model is utilized in order to convert PLNFR into DNFR within the secondary transformation. In terms of non-functionality requirements, the commonality as well as variability amongst the products is exploited through this transformation. For providing domain analysis within SPLE, the DNFR has provided efficient results.

Varun Gupta, et al. [19] proposed a novel non-functional prioritization mechanism in which the weighted average of stakeholder weight and respective preferences as per the parameters are utilized. There is change in priority of requirements when the usage count or dependencies are modified. Thus, each time when the priority is modified, the average change in priority is computed. Through the addition of average change in priority, the priorities of similar projects of modified and then as directly mapped requirement of new project, these similar projects are then assigned. As per the simulation results and comparisons made, it is seen that in terms of accuracy and computational time, the proposed approach provides better results in comparison to other existing approaches.

Jovan Popović, et al. [20] proposed computation of size of project on the basis of both functional as well as non-functional requirements

for effort estimation of Use Case Point method [20]. It is however not easy to assess the non-functional requirements due to which they are ignored. A methodology is proposed here on the basis of overall estimations, for assessing the nonfunctional requirements. In order to compare the correlation amongst unadjusted and adjusted use case weights, two data sets of the software projects are utilized. Thus, the guidelines required to develop software on the basis of effort estimation are attained through the results.

Klérison V. R. Paixão, et al. [21] presented a study related to the interplay amongst the nonfunctional requirements (NFRs). Here, by using 1283 software projects, the different statuses are generated. Amongst the NFR related generated statuses, several differences are identified. In order to enhance CI on the basis of efficiency and usability, the major focus is to remove the failures that arise, which is done by including tools. A bimodal distribution is identified along all the NFRs through time needed for placing a broken build back. On the basis of results evaluated it is seen that the delays related to broken builds are minimized through the functionalities and maintainability provided.

Rodrigo Veleza, et.al. [22] proposed an approach for reutilization of NFR knowledge that is stored in manner of solution patterns and is presented by the SIGs. In order to support the acquisition of knowledge for satisfying NFRs, the NDR tool is generated here as a support method on the basis of previously defined ontology. The several search capabilities offered by the NDR tool are presented in this work. Several levels of granularity along with numerous ways for searching and visualizing the results are utilized in order to provide the retrieval approach through which the existent knowledge is queried by the requirements engineer.

3. Proposed Methodology

The potential NFRs present within the system are identified within the initial phase of this analysis. On the basis of major assumptions of the cluster hypothesis, the research assumptions are generated within this phase. On the basis of relevance of information required [23], the documents that behave similarly are present within similar clusters. On the individual words of information content present in the documents, this behavior is attached. Thus, the grouping of words within similar cluster is done specifically [24]. On the basis of this approach, certain assumptions are used. The generation of semantically coherent groups of natural language words is the major objective of clustering the keywords that are extracted from the functional requirements of software system. Group of words $W = \{w_1, w_2, \dots, w_m\}$ into a set of disjoint groups of semantically similar words are clustered such that a $C = \{c_1, c_2, \dots, c_k\}$ is generated which is a set of disjoint groups that include semantically similar words. The conceptual themes that pass through the original text need to be described ideally by the clusters in C . the quality constraints of a system are mainly represented with the help of such themes. A NP-hard issue that arises here is the identification of optimal cluster configurations that are best fitted. However, with the help of certain experiments, the near-optimal solutions can be achieved [25]. The performances of series of semantic similarity measures as well as the word clustering techniques are evaluated in order to determine such configurations. Thus, from the FR words, the semantically coherent or thematic clusters can be generated.

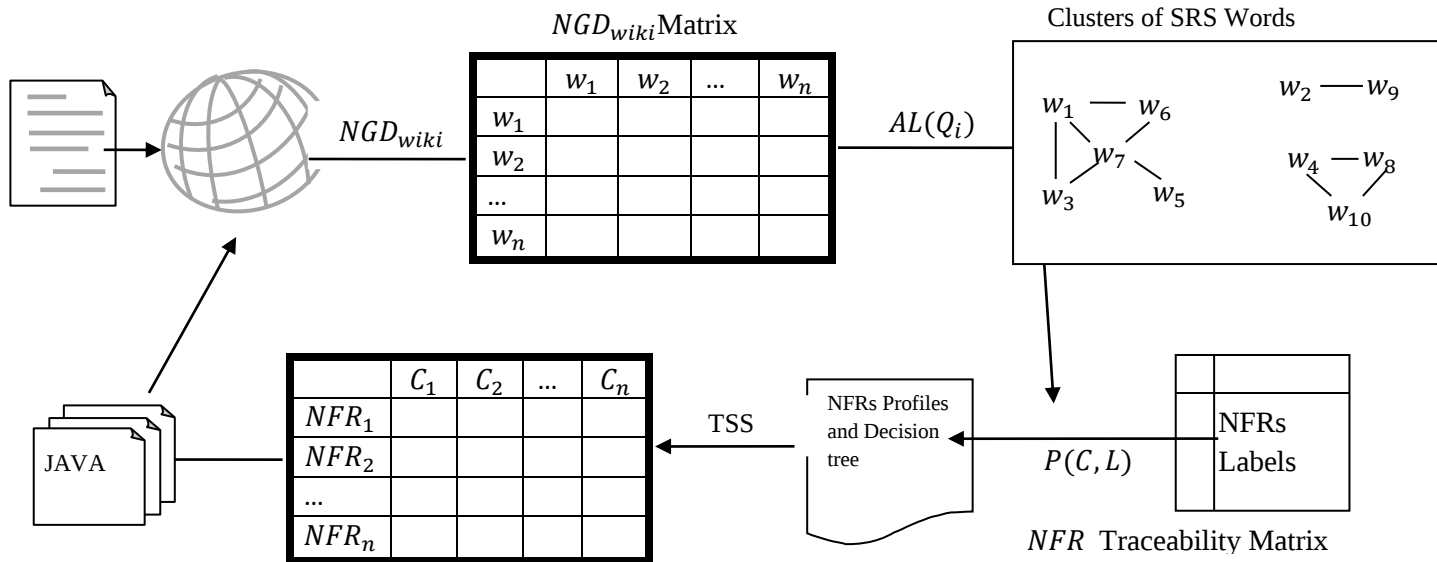


Fig. 1: NFR extraction and traceability

As shown in figure 1, the proposed approach is explained according to the steps followed. Here, NGD stands for Normalized Google Distance, AL stands for Average Linkage clustering algorithm, the cluster quality objective function is denoted by Q_i , the classification formula is given by $P(C, L)$, TSS stands for text semantic similarity as well as the code class is represented by C_i . In this research work the decision tree classifier is applied for the classification. These are the trees in which instances are classified by arranging them on the basis of feature values. A feature of an instance is represented by each node in a decision tree for the classification assumed value by the node is represented by each branch. These instances are classified at the root node initially after which on their feature values they are sorted. This learning has been utilized in data mining and machine learning. It is a predictive model in which item is observed on the map in order to identify item's target value. These tree models are also known as classification trees or regression trees. For the evaluation of decision trees, post-pruning techniques has been utilized as they are pruned by using a validation set.

4. Implementation and Results

This research work is based on the classification of null and non-null function from the IVR dataset. To do so, the NFR matrix is applied with the decision tree classifier for the data classification. The performance of proposed and existing method is compared in terms of accuracy and execution time. In the existing method only NFR matrix is applied for the data classification. In the proposed method, the decision tree is applied with NFR matrix for the classification. The proposed method show high accuracy and less execution time

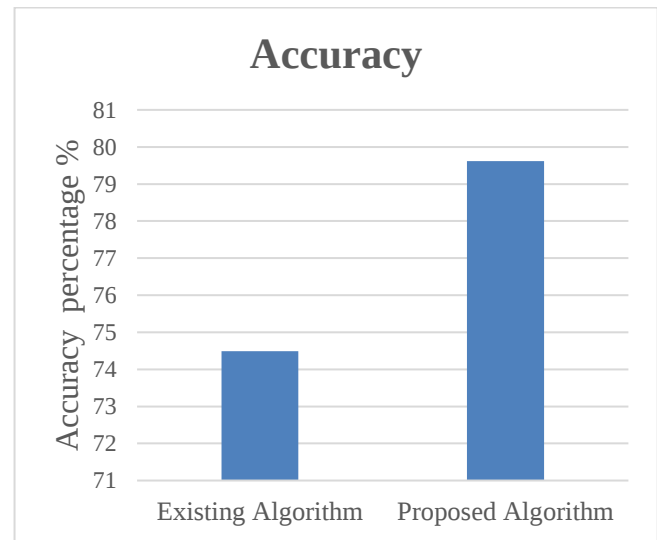


Fig. 2: Accuracy Comparison

As shown in figure 2, the accuracy of proposed method is compared with the existing method. It is analyzed that accuracy of proposed method is high as compared to existing due to use of decision tree with NFR matrix

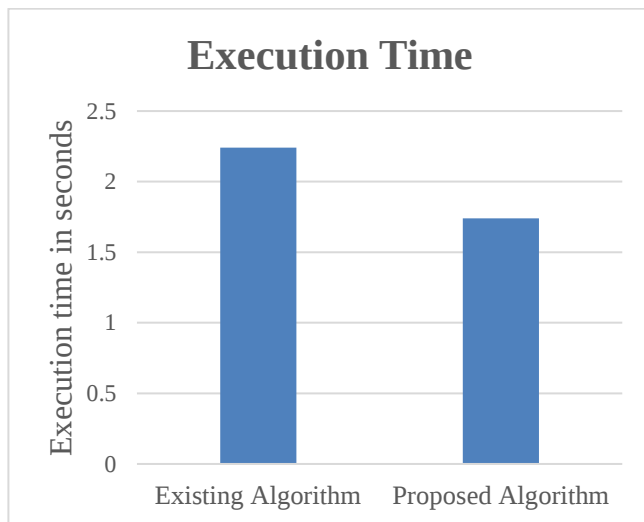


Fig. 3: Execution Time

As shown in figure 3, the execution time of proposed method is less as compared to existing method. It is due to use of NFR matrix with decision tree for the classification

5. Conclusion

In this work, it is concluded that null functions reduce the efficiency of the project. This research work is based on the classification of null functions from the project. The IVR dataset is considered for the performance analysis. The NFR matrix is used with the decision tree classification for the classification. The performance of proposed modal is tested in python and it shows high accuracy, less execution time when compared with existing technique.

References

- [1] R. Velela and L. M. Cysneiros, "An Initial Approach to Reuse Non-Functional Requirements Knowledge," in *Proceedings of the Eighth International i* Workshop (istar 2015)*, 2015, pp. 25–30.
- [2] P. P. Sancho, C. Juiz, R. Puigjaner, L. Chung, and N. Subramanian, "An Approach to Ontology-aided Performance Engineering Through NFR Framework," in *Proceedings of the 6th International Workshop on Software and Performance*, 2007, pp. 125–128.
- [3] M. M. Hasan, P. Loucopoulos, and M. Nikolaidou, "Classification and Qualitative Analysis of Non-Functional Requirements Approaches," Springer Berlin Heidelberg, 2014, pp. 348–362.
- [4] H. Hu, Q. Ma, T. Zhang, Y. Tan, H. Xiang, C. Fu, and Y. Feng, "Semantic modelling and automated reasoning of nonfunctional requirement conflicts in the context of softgoal interdependencies," *IET Softw.*, vol. 9, no. 6, pp. 145–156, 2015.
- [5] J. Eckhardt, A. Vogelsang, and D. M. Fernández, "Are 'non-functional' requirements really non-functional?: An investigation of non-functional requirements in practice," in *Proc. ICSE*, 2016, pp. 832–842.
- [6] L. Chung, B. A. Nixon, E. Yu, and J. Mylopoulos, *Non-functional requirements in software engineering*. Springer, 2012, vol. 5.
- [7] Cleland-Huang J, Heimdahl M, Huffman-Hayes J, Lutz R, Ma'nder P (2012) Trace queries for safety requirements in high assurance systems. In: *International conference on requirements engineering: foundation for software quality*, pp 179–193
- [8] Mahmoud A (2015) An information theoretic approach for extracting and tracing non-functional requirements. In: *International requirements engineering conference*
- [9] Mihalcea R, Corley C, Strapparava C (2006) Corpus-based and knowledge-based measures of text semantic similarity. In: *National conference on artificial intelligence*, pp 775–780, 2006.
- [10] Holzinger A, Yildirim P, Geier M, Simonik KM (2013) Quality-based knowledge discovery from medical text on the web. In: Pasi G, Bordogna G, Jain L (eds) *Quality issues in the management of web information*. Springer, Berlin, pp 145–158
- [11] Huffman-Hayes J, Dekhtyar A, Sundaram S (2006) Advancing candidate link generation for requirements tracing: the study of methods. *IEEE Trans Softw Eng* 32(1):4–19
- [12] Mahmoud A, Niu N (2015) On the role of semantics in automated requirements tracing. *Requir Eng* 20(3):281–300
- [13] Lo D, Nagappan N, Zimmermann T (2015) How practitioners perceive the relevance of software engineering research. In: *Joint meeting on foundations of software engineering*, pp 415–425
- [14] Arthur D, Vassilvitskii S (2007) K-means++: the advantages of careful seeding. In: *Annual ACM-SIAM symposium on discrete algorithms*, pp 1027–1035
- [15] Deisenbock F, Pizka M (2005) Concise and consistent naming. In: *International workshop on program comprehension*, pp 97–106
- [16] Bekkerman R, El-Yaniv R, Tishby N, Winter Y (2003) Distributional word clusters vs. words for text categorization. *J Mach Learn Res* 3:1183–1208
- [17] Deerwester S, Dumais S, Furnas G, Landauer T, Harshman R (1990) Indexing by latent semantic analysis. *J Am Soc Inf Sci* 41(6):391–407
- [18] Fatima Khalique, Wasi Haider Butt, Shoab Ahmad Khan, "Creating Domain Non-Functional Requirements in Software Product Line Engineering using Model Transformations", 2017 International Conference on Frontiers of Information Technology
- [19] Varun Gupta, Mohd. Talha Khan, Chetna Gupta, "Non-Functional Requirement Prioritization Approach Based on Historical Similar Project", 2017, IEEE
- [20] Jovan Popović, Nenad Korolija, Željko Marković, and Dragan Bojić, "The Influence of Non-functional Requirements in UCP Method on the Accuracy of Effort Estimates", 2017, 25th Telecommunications forum TELFOR
- [21] Klérison V. R. Paixão, Crícia Z. Felício, Fernanda M. Delfim and Marcelo de A. Maia, "On the Interplay between Non-Functional Requirements and Builds on Continuous Integration", 2017 IEEE/ACM 14th International Conference on Mining Software Repositories (MSR)
- [22] Rodrigo Velela, Luiz Marcio Cysneiros, "Towards a Tool to Help Exploring Existing Non-Functional Requirements Solution Patterns", 2017 IEEE 25th International Requirements Engineering Conference Workshops
- [23] Niu N, Mahmoud A (2012) Enhancing candidate link generation for requirements tracing: the cluster hypothesis revisited. In: *IEEE international requirements engineering conference*, pp 81–90
- [24] Hearst M, Pedersen J (1996) Reexamining the cluster hypothesis: scatter/gather on retrieval results. In: *International ACM SIGIR conference on Research and development in information retrieval*, pp 76–84
- [25] Anas Mahmoud, Grant Williams, "Detecting, classifying, and tracing non-functional software requirements", Springer-Verlag London 2016