Performance Evaluation of Requirements Engineering Methodology for Automated Detection of Non Functional Requirements

J.Selvakumar
Assistant Professor in Department of Software Engineering (PG)
Sri Ramakrishna Engineering College
Coimbatore-22, India
selvakumar_me@yahoo.co.in

Dr. M. Rajaram
Anna University of Technology
Tirunelveli, India

***Abstract*—Requirement Engineering (RE) deals with the requirements of a proposed solution and handles conflicting requirements of the various stakeholders and is critical to the success of a project. Good requirement engineering methodologies should be measurable, testable and should be sufficient for system design. Another important aspect of Requirement engineering is to capture the non functional requirements which have to be considered early to avoid system level constraints, security issues and overall quality issues. Typical Non Functional Requirements (NFR) are identified in both structured as well as unstructured documents. With availability of automated tools for requirement tracing and identifying NFRs, one method to find the effectiveness of the requirement engineering methodology in automation is its capability to capture NFR in an effective manner. In this paper we investigate the effectiveness of Information Retrieval (IR) methods for identifying NFRs.**

Keywords-Requirement Engineering, Functional Requirement, Non Functional Requirement, Information Retrieval, Bagging.

I. INTRODUCTION

Requirement Engineering(RE) for Software Requirement Specification (SRS) includes all activities including discovery, validating, documenting and managing requirements[1]. The quality of RE has been found to be related directly with the final software quality[2].Various research has shown there exists a direct relationship between the quality of requirements and the density of defects in software[3]. The RE methodology and process is part of RE and specifies how the requirements has to be gathered[4]. For successful Software Development Life Cycle (SDLC), the RE process must be improved when they fail to meet their desired purposes. This leads to finding methods to measure the RE process quantitatively and apply improvements against the measured deficiency in the RE process.

Requirements can be broadly classified into Functional Requirements (FR) and Non Functional Requirements(NFR). FR deals with requirements that affect the functionality of the system whereas NFR deals with requirements that constrain the system[5]. For most part when requirements is spoken of, it refers to functional requirements characterized by

- Simple Language
- specific to a business requirement
- Describes what and not how.

Since NFR identifies user or system constraints it is characterized by features such as[6]

- User-friendliness

- Response time
- Portability
- Reliability
- Maintainability.

Figure I illustrates the steps to identify NFR

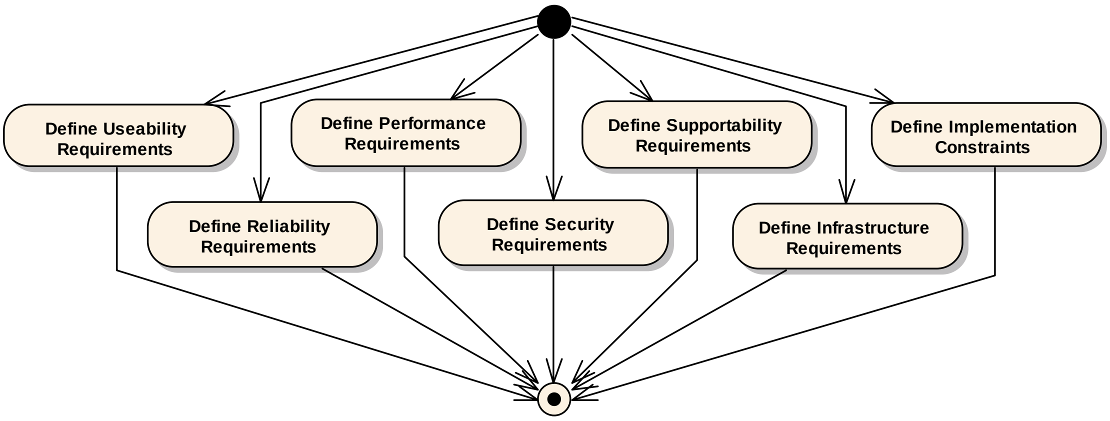

Figure I : Steps to capture Non Functional Requirements

It is seen from figure I that NFRs are also part of FR and may appear regularly when FR is being elicited. Since NFR are part of FR its discovery is sometimes missed out during the initial stages of the SDLC. The different views of stakeholders on a NFR will also avoid clarity on the system wide NFR.

Tracing FR and NFR is a time consuming task and requires experienced analyst to identify NFR hidden in business requirements. Various methods have been proposed using data mining methodologies to automate SDLC with the goal of decreasing labor intensive tasks and speed up the performance[7]. Data mining tasks in requirement engineering could fall into the following categories [8,9,10,11]

- Predicting labels using mining algorithms based on training data provided with labels.
- Frequency and Pattern mining
- Clustering requirements based on their closeness with each other.

In this paper we investigate prediction algorithm to identify NFR from the requirements document. This paper is organized into the following sections. Section II describes the proposed methodology, Section III discusses the result obtained and section IV concludes this work.

II. PROPOSED METHODOLOGY

The goal of this work is to evaluate the performance of a classifier in predicting the class of NFR associated with the requirement document. To validate the classifier we use the NFR dataset available in the promise data repository[12]. The NFR dataset consists of 15 requirement specifications of MS student projects and has a total of 326 NFRs and 358 FRs. The NFR categories included availability, scalability, usability, security. One NFR category in the dataset was portability and since only one class label existed, the instance was removed from this study. Features were extracted from each requirement document using the word occurrence criteria. Bagging and boosting methods were investigated on the extracted data.

Bagging[12] based classifiers generated multiple versions of predictors and using the same to get an aggregated predictor. A predictor $\varphi(x,l)$ predicts a class label $j \in \{1,2,...,J\}$. With the learning set l, Φ predicts class label *j* at input *x* with relative frequency

$$Q(j\,|\,x) = P(\phi(x,l) = j) \qquad (1)$$

The probability that the predictor classifies x correctly is

$$\sum_j Q(j\,|\,x)P(j\,|\,x) \qquad (2)$$

Boosting[13] works by using classification algorithms sequentially on the reweighted versions of the training data. The final class label predicted is based on the weighted majority vote. In logitboost the initial weights is set at 1/N where N is the number of instance with the probability estimate p(x_i=0.5). The process is repeated m times and the function is fitted using least squares regression.

III. Results and Discussion

The precision and recall for both the classifiers for the predicted class labels is shown in table I.

Table I: Precision and Recall on the investigated data.

	Logitboost		Bagging	
	Precision	Recall	Precision	Recall
Performance	0.83	0.722	0.844	0.704
Look and Feel	0.231	0.079	0.625	0.132
Usability	0.569	0.433	0.414	0.433
Availability	0.8	0.571	0.727	0.381
Security	0.588	0.455	0.578	0.394
Functional	0.625	0.894	0.608	0.906
Fault Tolerance	0	0	0.333	0.1
Scalability	0.333	0.333	0.25	0.095
Operational	0.543	0.403	0.549	0.452
Legal	0.333	0.077	0	0
Maintainability	0.176	0.013	0.667	0.118

From Table I it is seen that the performance variation of class label fault tolerance, legal, and look and feel between the two classifiers are very high. However the other class have good retrieval accuracy. The classification accuracy of the classifier is shown in Figure II.

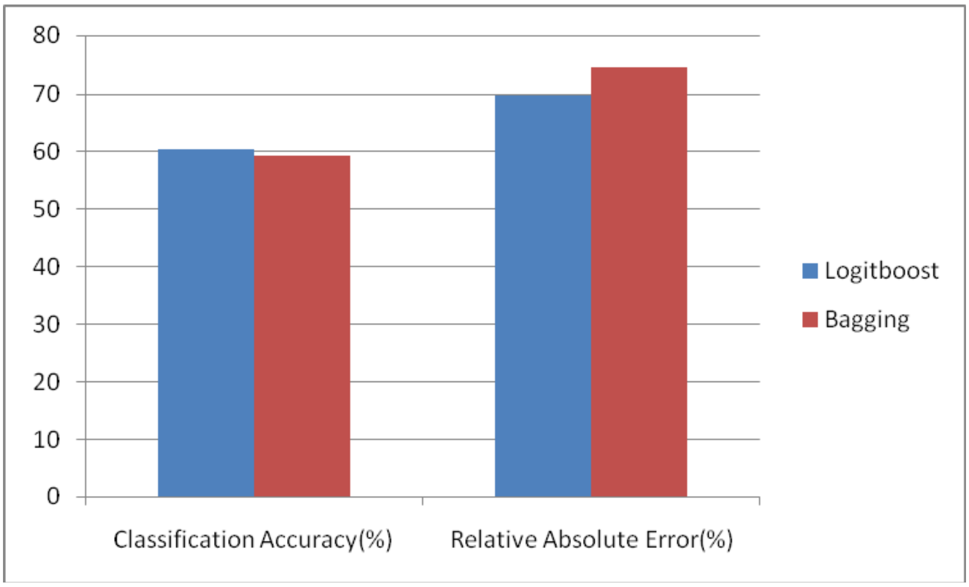

Figure II : Classification Accuracy and Route Absolute Error.

From figure II it is seen that using a simple method as word frequency to find out the non functional requirements produces good results. The results obtained may not be sufficient for an automated system, but will find usefulness to the analyst manually identifying the NFRs. It is also seen that textual requirement engineering may not give very high accuracy results for mining NFR, however further investigation need to be carried out.

IV. CONCLUSION

In this work we investigate data mining approach to identify Non Functional Requirements (NFR) from Functional Requirement (FR) Documents. In the proposed method we use public dataset available in the promise database repository and investigate Bagging and Logitboost classification algorithms. 57 words based on their importance were extracted from the requirement document for the data mining operation. Results obtained were satisfactory. Further work needs to be done by using Neural Network based retrieval system and preprocessing the data with Singular Value Decomposition. Multimedia based requirements engineering also needs to be investigated to identify the efficiency of information retrieval systems.

REFERENCES

[1] L. A. Macaulay, Requirements Engineering, vol. 1, First ed. Manchester, UK: Springer-Verlag, 1996.
[2] J. Voas, "Assuring Software Quality Assurance,"IEEE Software, vol. 20(3), pp. 48-49, 2003.
[3] K. Wiegers, "The Real World of Requirements Engineering," Requirenautics Quarterly, vol. The Newsletter of the BCS Requirements Engineering, pp. 5-8, 2004.
[4] M. K. Niazi, "Improving the Requirements Engineering Process Through the Application of a Key Process Areas Approach," AWRE02, 2002.
[5] J. Li, A. Eberlein, and B. H. Far, " Evaluating the Requirements Engineering Process using Major Concerns",Software Engineering, vol. 418, pp. 237-252, 2004.
[6] http://www.cragsystems.co.uk/development_process/gather_non_functional_requirements.htm
[7] Jane Huffman Hayes and Alex Dekhtyar and Senthil Sundaram " Text Mining for Software Engineering: How Analyst Feedback Impacts Final Results"SIGSOFT Softw. Eng. Notes, Vol. 30, No. 4. (2005), pp. 1-5.
[8] Tao Xie,Jian Pei, " MAPO: mining API usages from open source repositories" Proceedings of the 2006 international workshop on Mining software repositories, pp:54-57
[9] Vinod Ganapathy,"Mining securitysensitive operations in legacy code using concept analysis",Proceedings of the 29th International Conference on Software Engineering ,2007.
[10] Andy Podgurski, David Leon, Patrick Francis, Wes Masfi, Melinda Minch. " Automated support for classifying software failure reports", ICSE, 2003, pp: 465-475
[11] Américo Sampaio, Neil Loughran, Awais Rashid , Paul Rayson ."Mining Aspects in Requirements" In Workshop on Early Aspects, 2005, pp:15
[12] http://promisedata.org/?p=38
[13] Jerome Friedman, Trevor Hastie and Robert Tibshirani "Additive Logistic Regression : A Statistical View of Boosting" Annals of statistcs, Vol 28, No.,2, 2000, pp:337-407

AUTHORS PROFILE

J.Selvakumar received the B.E degree in Computer Science & Engineering from the Madras University in May 2001 and the M.E degree in Computer Science & Engineering from the Bharathiyar University in December 2002. He is currently working towards the Ph.D degree at the Anna University, Chennai . He is currently an Assistant Professor. His research interest is Requirements Engineering.

Dr.M.Rajaram received the B.E. Degree in Electrical and Electronics Engineering from Alagappa Chettiar College of Engineering and Technology, Karaikudi in 1981, M.E. Degree in Power Systems at Government College of Technology, Coimbatore, in 1988 and Ph.D in Control Systems at PSG College of Technology, Coimbatore in 1994.He is currently the Vice Chancellor of Anna University of Technology, Tirunelveli, India.

Being one among the Senior Professors and, having completed 29 years of Teaching in various Government Engineering Colleges at Tirunelveli, Coimbatore, Karaikudi, Salem and Vellore in Electrical and Electronics Engineering Discipline, he has successfully supervised 11 Ph.D and M.S (by Research) Scholars who are awarded with the Doctoral Degree and 10 candidates are pursuing Ph.D under his guide ship at present.

To his record he has published his Research findings in 75 International Journal/National Journals, 61 International Conferences and 53 National Conferences. He has also authored 6 Text Books.