

Supervised Machine Learning Techniques to Classify Software
Requirements



Submitted by: Faiza Latif ✓
Registration No: 01-241161-005
Supervisor Name: Dr. Muhammad Asfand-e-Yar ✓

A thesis submitted to the Department of Software Engineering, Bahria University, Islamabad in the partial fulfillment for the requirements of a Master's degree in Software Engineering.

March, 2018

Abstract

Requirements are the functionalities that are discovered before building any product. A systematic approach through which the software engineer collects requirements from diverse sources and implements them into the software development processes is called Requirement engineering. Requirement engineering contains a set of activities for discovering, analyzing, documenting, validating and maintaining a set of requirements for a system. Functional Requirements (FRs) and Non-Functional Requirements (NFRs) are two basic types of requirements in Software Requirement Specification documents. The classification of these requirements is an important task as it provides an ease for the team manager and the software development first. The NFRs grab less attention from the development team. Some of the NFR categories are very important to consider while developing the software. This research study proposes a technique to classify the requirements into FRs and NFRs with the help of Machine Learning techniques. The NFRs are defined in the requirement document but in some cases the NFRs are not clearly mentioned. Therefore, using Machine Learning the requirements are classified and system will automatically identify the categories of NFRs, which are evaluated using accuracy, precision and F-Measure. The accuracy explains that whether the NFR is accurately classified in the document, the precision explains whether the mentioned requirements are properly placed in the classified field. F-measure or F-score is the weighted average of precision and recall. Furthermore, it also classifies the NFRs into sub categories. Different ML approaches and classification algorithms will be used in the study.

Contents

Chapter 1: Introduction	11
1.1 Software Requirement specification	1
1.1.1 Functional Requirements.....	1
1.1.2 Non-functional Requirements	1
1.1.3 Non-Functional Requirements types.....	2
1.2 Machine learning.	2
1.2.1 Supervised machine learning	3
1.2.2 Unsupervised machine learning	3
1.3 Motivation.....	3
1.4 Problem Statement	4
1.5 Research Questions.....	4
1.6 Research Methodology	5
1.7 Thesis Organization	5
Chapter 2: Software Requirements Classification – Methods in Literature.....	6
2.1 Software requirements classification	6
2.2 Importance of classification.....	6
2.3 Machine Learning	7
2.4 Approaches for SRS classification in literature.....	7
2.5 Supervised Text Classification Methods	10
2.5.1 K-Nearest Neighbor (KNN).....	10
2.5.2 Naive Bayes Classifier	11
2.5.3 Decision Tree Classifier.....	11
2.5.4 Support Vector Machine	11
2.6 Feature Extraction and selection.....	11
2.6.1 Feature subset Selection	12
2.6.1.1 Feature Subset Selection Process	13
2.6.1.2 Subset Generation	13
2.6.1.3 Subset Evaluation.....	14
2.6.1.4 Stopping Criteria.....	14

2.6.1.5	Result Validation	14
2.6.1.6	Feature Subset Evaluation Techniques	15
2.6.1.6.1	Filters	15
2.6.1.6.2	Wrappers	16
2.7	Application of Feature Selection in text categorization.....	16
2.8	Class balancing	17
Chapter 3	Proposed Methodology	18
3.1	Dataset Acquisition	19
3.2	Document Preprocessing.....	20
3.2.1	Tokenization	21
3.2.2	Stop words removal	22
3.2.3	Stemming	23
3.3	LDA Models.....	24
3.4	Feature Extraction	25
3.5	Class balancing.....	26
3.6	Classification algorithms.....	28
3.6.1.	Random Forest	28
3.6.2	Decision Tree	29
3.6.3	Gradient Boosting	29
3.6.4	Extreme Gradient Boosting.....	29
Chapter 4:	Experimental Results and Evaluation.....	30
4.1	Feature Extraction and Selection Process	30
4.1.1	Before Class Balancing.....	31
4.1.2	After Class Balancing	31
4.2	Class Balancing	32
4.2.1	Before Class Balancing.....	32
4.2.2	After Class Balancing	35
4.3	Classifier Performance	37
4.3.1	Gradient Boosting	37
4.3.2	Decision Tree	38

4.3.3	Random Forest.....	39
4.3.4	Extreme Gradient Boosting.....	40
Chapter 5	Conclusion and Future Work	43
References.....		45