DEFECT SEVERITY PREDICTION USING TEXT MINING

A Dissertation submitted in the partial fulfillment for the award of

MASTER OF TECHNOLOGY

IN

SOFTWARE TECHNOLOGY

by

Yogesh Kumar Srivastava

Roll No: 2K11/ST/20

Under the Guidance of

Dr. Ruchika Malhotra

Assistant Professor

Department of Software Engineering

Department of Software Engineering

Delhi Technological University

New Delhi

2014

DECLARATION

I hereby declare that the thesis entitled "**DEFECT SEVERITY PREDICTION USING TEXT MINING"** which is being submitted to the **Delhi Technological University**, in partial fulfillment of the requirements for the award of degree of **Master of Technology in Software Technology** is an authentic work carried out by me. The material contained in this thesis has not been submitted to any university or institution for the award of any degree.

Yogesh Kumar Srivastava

Department of Software Engineering

Delhi Technological University,

Delhi.

CERTIFICATE

DELHI TECHNOLOGICAL UNIVERSITY

Date: _____

This is to certify that the thesis entitled **"Defect Severity Prediction Using Text Mining"** submitted by **Yogesh Kumar Srivastava (Roll Number: 2K11/ST/20),** in partial fulfillment of the requirements for the award of degree of Master of Technology in Software Engineering, is an authentic work carried out by her under my guidance. The content embodied in this thesis has not been submitted by her earlier to any institution or organization for any degree or diploma to the best of my knowledge and belief.

Project Guide

Dr. Ruchika Malhotra

Assistant Professor

Department of Software Engineering

Delhi Technological University, Delhi-110042

ACKNOWLEDGEMENT

I take this opportunity to express my deepest gratitude and appreciation to all those who have helped me directly or indirectly towards the successful completion of this thesis.

Foremost, I would like to express my sincere gratitude to my guide **Dr. Ruchika Malhotra**, **Assistant Professor, Department of Software Engineering**, **Delhi Technological University, Delhi** whose benevolent guidance, constant support, encouragement and valuable suggestions throughout the course of my work helped me successfully complete this thesis. Without her continuous support and interest, this thesis would not have been the same as presented here.

Besides my guide, I would like to thank the entire teaching and non-teaching staff in the Department of Software Engineering, DTU for all their help during my course of work.

YOGESH KUMAR SRIVASTAVA

2K11/ST/09

Master of Technology (Software Technology)

Delhi Technological University

Bawana road, Delhi - 110042

Table of Contents

INTRODUCTION 1

1.1 Motivation of the Work 2

1.2 Goals of the thesis 4

1.3 Organization of the Thesis 5

LITERATURE SURVEY 7

TEXT MINING 14

3.1 Tokenization: 14

3.2 Stop word removal: 15

3.3 Stemming: 16

3.4 Tf*idf: 16

3.5 InfoGain: 17

PROPOSED FRAMEWORK FOR DEFECT SEVERITY PREDICTION 19

EMPIRICAL STUDY 22

DATA COLLECTION & EXPERIMENT DESIGN 25

6.1 Objects of Experiment Design 25

6.2 Tool for standard Text Mining Techniques 26

6.3 Tool for Machine Learning Techniques 32

RESULTS ANALYSIS 33

7.1 Test data 1 33

7.1.1 Result Summary 34

7.2 Test Data 2 (PITS C) 38

7.2.1 Result Summary 38

CONCLUSION & FUTURE WORK 45

REFERENCES 47

List of Figures

Figure 1. Defect Introduction....9
Figure 2. Defect removal....11
Figure 3. Generality of Basic Text Mining Technique....15
Figure 4. 24 of the 262 stop words used....16
Figure 5. Process flow of Severity Prediction for incoming Defect Report....24
Figure 6. Initial Screen....26
Figure 7. After the file has been selected....27
Figure 8. After the File has been loaded....28
Figure 9. After Stop word removal....29
Figure 10. After Stemming....30
Figure 11. After applying InfoGain....31
Figure 12. ROC Curve corresponding to Top 5 Dataset – Test Data 1....34
Figure 13. ROC Curve corresponding to Top 10 Dataset – Test Data 1....35
Figure 14. ROC Curve corresponding to Top 20 Dataset – Test Data 1....36
Figure 15. ROC Curve corresponding to Top 30 Dataset – Test Data 1....37
Figure 16. ROC Curve corresponding to Top 50 Dataset – Test Data 1....38
Figure 17. ROC Curve Corresponding to Top 5 Dataset – Test Data 2 (PITS C)....39
Figure 18. ROC Curve Corresponding to Top 10 Dataset – Test Data 2 (PITS C)....40
Figure 19. ROC Curve Corresponding to Top 25 Dataset – Test Data 2 (PITS C)....41
Figure 20. ROC Curve Corresponding to Top 50 Dataset – Test Data 2 (PITS C)....42
Figure 21. ROC Curve Corresponding to Top 75 Dataset – Test Data 2 (PITS C)....43

Figure 22. ROC Curve Corresponding to Top 100 Dataset – Test Data 2 (PITS C).....44

List of Tables

Table 1. Severities for Robotic Missions 12

Table 2. Confusion Matrix 20

Table 3. Result Summary with Top 5 Dataset for Test Data 1 34

Table 4. Result Summary with Top 10 Dataset for Test Data 1 35

Table 5. Result Summary with Top 20 Dataset for Test Data 1 35

Table 6. Result Summary with Top 30 Dataset for Test Data 1 36

Table 7. Result Summary with Top 50 Dataset for Test Data 1 37

Table 8. Result Summary with Top 5 Dataset for Test Data 2 38

Table 9. Result Summary with Top 10 Dataset for Test Data 2 39

Table 10. Result Summary with Top 25 Dataset for Test Data 2 40

Table 11. Result Summary with Top 50 Dataset for Test Data 2 41

Table 12. Result Summary with Top 75 Dataset for Test Data 2 42

Table 13. Result Summary with Top 100 Dataset for Test Data 2 43

ABSTRACT

The objective of this thesis is to help in predicting the defects severity automatically. There are databases which are used for logging the defects during the testing phases. It is quite possible that one defect database which is working fine for one system or project may not work fine for the other project. It may also happen the one defect database may not work for multiple projects. Moreover the defects data get collected for a project lack in consistency. Although all projects are having a predefined set of data fields which were required but these data fields do not provide enough information where quality of the issues can be found and we can compare the projects. The main purpose of this project is to first develop a tool which helps in predicting the defect severity in testing process.

Here training data is taken in the text file. It is important to reduce the unnecessary words and get the set of words by which the text classification can be done properly. There are lots of text mining techniques available which can be used to reduce the unnecessary words or we can say that words which are not helpful in the classification can be removed. There are lots of common words which are useless for classification can be removed and it is called as stop word removal. Before creating bag of words all occurrence of these types of words can be removed by stop word removal.

Even with the Stop Word removal [5] it is not possible to get the required set of word which can be used for classification so reduce it further can be achieved by Stemming. Number of words present is document get analyzed in case of Stemming. The purpose of stemming to find out the set of words which can be treated as similar or equivalent words. Like we can say

'applied', 'applying', 'applies' and 'apply' are similar words. After stemming the term frequency timed inverse document frequency is calculated which is often denoted as "tf*idf". To simplify the target, InfoGain is applied to word based. After applying InfoGain these words can be used for classification. On these training data then we can apply the machine language so that it can learn the rules to predict the severity by finding the terms.

Chapter 1

INTRODUCTION

Text data mining which is most commonly called as Text mining. Text mining is form of text analytics which is generally getting relevant and meaningful information from the text. Through devising of patterns and trends we can typically derive the high-quality information by using statistical pattern learning. When we structured any given text file and parse with additional linguistic features and removal of unnecessary words and creating any pattern where we can able to evaluate the data are the steps involved in the text mining. Relevant and meaningful information means that data can be used to infer any information. Text mining is learning relations between named entities which can be said that text mining tasks include categorization of texts, its clustering, extraction of any information, its analysis, summarization and defining relationship.

In case of text mining we can analyze the texts from which we can retrieve the information, can be used for recognizing the pattern, extracting the information and data can be used for prediction. The idea is from the given text file extracting the relevant and meaningful texts and reaching the level where these texts can be used for further analysis which can result in inferring the relevant information.

Usually if we have any text mining application then it has checks set of documents written in any language may be natural language and we can generate some model which can be used for extracting the information or can be used as a model. A set of machine learning

techniques which can be modeled and structured the content of the information for textual sources describes the term text analytics which can be used for further data analysis or research.

We can say that the term "text analytics" can also be understand as the analysis of the text which can be a response to a business problem, whether it can solve the problem independently or along with queries, analysis of fields and numerical data. As per current analysis and understanding around 80 percent of the information is in unstructured form.

Here text mining is used to predict the severity of the defects automatically. We seek to monitor any project using the database for defect logging during testing phases. For training data set a defect database is used which is in the form of text file and after filtering the information the training data is used for classification. Based on this classification the severity of defects gets predicted automatically. The idea is to have uniform classification of the defects.

1.1 Motivation of the Work

In software development life cycle, severity prediction on defect reports is big problem obtaining research attention due to the considerable triaging cost. Several text mining approaches have been proposed to predict the severity using different advanced learning models during several research works in the past. Different text mining approaches demonstrate the effectiveness of predicting the severity of defects in the software life cycle. In this project I discuss whether feature selection can benefit the severity prediction task with Information Gain.

In software maintenance debugging is major concern because it generally costs more efforts and time for making the correction. If we are having large scale software projects with huge

numbers of defect report, then the debugging cost increases exponentially because of the huge amount of debugging cost. To save the cost with respect to time and efforts it is important to effectively utilize the limited resources in processing the defects which are getting logged during the testing phases. It is important to analyze that which defect report or defects need to be focused more urgently. If we do not have the historical data and expertise then estimation methods such as analogy and planning becomes unpredictable. To overcome this type of situation it is important to have some simple and strong algorithm in place which plays an important role in affecting the size, cost and duration of any project. This algorithm also provides the basic ground for the people who are not having much experience to estimate the project more precisely. Unless and until we have the proper estimation we cannot plan for the project effectively and end up increasing the project cost and efforts.

The need for defect severity prediction is important to have uniformity and updating the developer for the criticality of the issue. Based on this information the stakeholder can take proper and clear decision.

It is important because of following reasons:

- ***Testing shows presence of errors.*** This implies that one cannot be assured that software is free from errors. It shows errors are present but cannot assure their absence.
- ***Testing depends on context***. No two systems are the same and therefore, cannot be tested the same way. Testing intensity, when to stop testing etc. must be defined individually for each system depending on its testing context.
- ***False conclusion: no errors equal usable system***. Error detection and removal does not guarantee a usable system matching the user's expectations. Early integration of units and rapid prototyping prevents unhappy clients and discussions**.**

1.2 Goals of the thesis

The goal of the work in this thesis is summarized below:

To give insight the mechanism to predict the severity of the defects automatically, we have to monitor any project which is using the any database either commercialized or in-house developed for defect logging during software testing phases. If we apply any type of queries analyze the defect database then it may works well for this project, but it is quite possible that it may not work well for other or multiple projects. Also, the way project collects defect data is not consistent. In defect database there are set of data fields which are required but if we see then the majority of these fields do not help in inferring the quality of the issue and we cannot use these fields to compare the multiple projects. In this it is important to find out a solution which can be used to predict the severity of defects more efficiently.

Here training data is taken in the text file. It is important to reduce the unnecessary words and get the set of words by which the text classification can be done properly. There are lots of text mining techniques available which can be used to reduce the unnecessary words or we can say that words which are not helpful in the classification can be removed. There are lots of common words which are useless for classification can be removed and it is called as stop word removal. Before creating bag of words all occurrence of these types of words can be removed by stop word removal.

Even with the Stop Word removal it is not possible to get the required set of word which can be used for classification so reduce it further we have to use Stemming. In Stemming technique the number of words present is document get analyzed. The purpose of stemming to find out the set of words which can be treated as similar or equivalent words. After stemming the term frequency timed inverse document frequency is calculated which is often denoted as "tf*idf". To simplify the target, InfoGain is applied to word based. After applying

InfoGain these words can be used for classification. On these training data then we can apply the machine language so that it can learn the rules to predict the severity by finding the terms.

1.3 Organization of the Thesis

Organization of this Thesis is as follows:

- ***Chapter 2*** explains the previous work done in the field of Defect Severity Prediction. This includes the extensive study of PITS system developed for NASA to automatically predict the severity of the defects. It also highlights some of the most relevant works in the direction of field of work presented in the thesis.

- ***Chapter 3*** gives a comprehensive study of Text Mining. This chapter is dedicated to a profound study of historical background of Text Mining including details of its origin, various phases of Text Mining, significance and utility of the algorithm. We also exemplified the working of the algorithm with some sample data.

- ***Chapter 4*** focuses on the proposed framework for Defect Severity Prediction problem including the details of the framework. It also describes the Tools used to take the test data and getting the test data after applying various Text Mining techniques.

- ***Chapter 5*** presents the research questions that we aim to address in this thesis. We also describe in detail the Defect Severity Prediction framework.

- ***Chapter 6*** comprises of the empirical data collection for five test results for mobile phone testing. It also includes the details of the experiments and how the experiments were performed.

- ***Chapter 7*** presents a detailed analysis of the results obtained. In this chapter we compare and assess the results of the experiments by the tool developed for Defect Severity Prediction using Text Mining.

- ***Chapter 8*** presents the conclusions of the thesis and future work.

Chapter 2

LITERATURE SURVEY

Two papers have taken for study and laying down the basic concept of this project. These papers are:

- "Improving IV&V Techniques Through the Analysis of Project Anomalies: LINKER - preliminary report"[2]
- "Automated Severity Assessment of Software Defect Reports"[1]

These papers emphasize on the importance of importance of recognizing properly the severity of issues identified by Test Engineers during the testing phase. These papers give a study which was done for mission critical system developed by NASA. It is essential that proper severity assessment is done so that appropriate resources can be allocated, proper planning can be done for fixing the defects and additional testing can be planned. Generally in testing phase the assessment of defects is usually based test engineers and test engineers can be give this assessment based on their experience and time taken by test engineers to detect the defect.

These papers talks about a method SEVERIS (SEVERity ISsue assessment) [2], which helps in assigning uniform severity levels to defects which got reported. SEVERIS system is based on the concept of text mining and machine learning techniques. Using these techniques we can evaluate the collected defects information in the defects report.

This paper also emphasize on the usage of the SEVERIS system which runs on the data collected from the PITS (Projects from their Issue Tracking System) from the NASA projects. The results of this paper claim that SEVERIS can be good tool for defining or assigning severity to defects automatically, which can be easily and effectively used by the test engineers.

When this project was developed then it got divided in two parts which are described in above two papers which is used for the bases in this project. The main purpose of first paper is to get the experience with different data sources available, like [1] & [2]

1. SILAP, from the IV&V planning and scoping team
2. James Dabney's Bayes networks that describe the IV&V business practices of the L3 IV&V contractor
3. The PITS issue tracking data
4. The LINKER database project that intends to join PITS to other data sources
5. Balanced score card strategy maps from NASA Langley
6. COCOMO data sets from JPL

And the conclusion from the above analysis is:

Initial report got generated which described that what had been learned. Based on the initial analysis ranking was assigned to the most common WBS [2]. WBS ranking indicates that which task will benefit most if it is optimized.

Initial analysis was generated after applying text mining [3] on the PITS database which resulted a system by which results can be concluded. This expert system was reviewed and updated based on the test engineer's proposed severity level.

This paper also describes a new database LINKER during IV&V [2] phase which collates and combines data generated from different sources. Till the paper was published LINKER was still under development. So we can say that, unlike the prior reports, we cannot get any report which reports experimental results. Rather, these papers discuss more about the benefits and drawbacks of LINKER database.

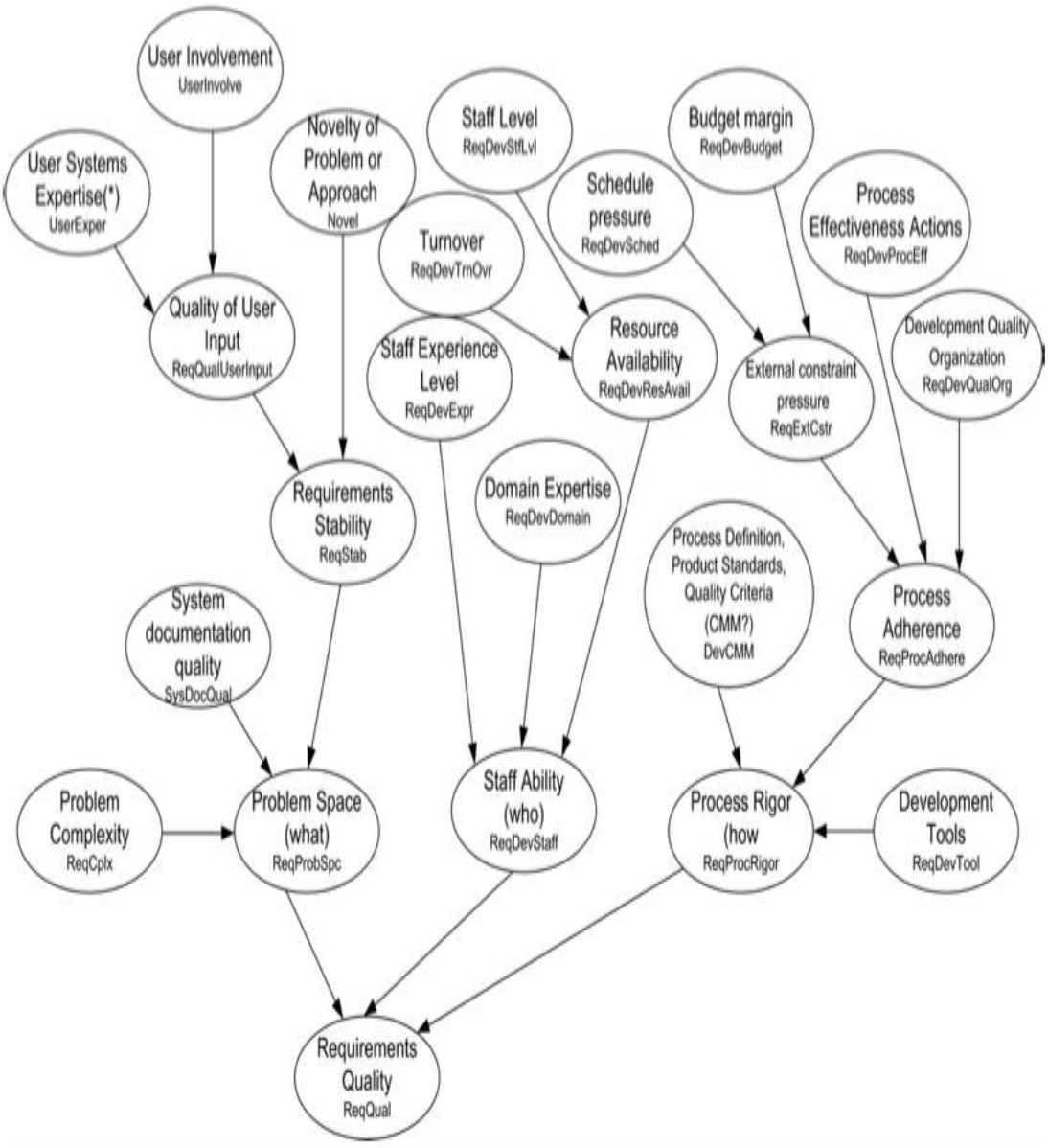

Figure 1. Defect Introduction

SEVERIS (SEVERity ISsue assessment) system was the automated method detailed in the second paper; this method helps the test engineer for assigning severity levels. PITS is a database which captures all the finding in NASA's software. To analyze this case study the data available in PITS got collected for many years and which includes all the issues have been reported in human-rated systems and robotic satellite missions [2].

Currently, there is lot or defect tracking systems available which are either commercially available or developed in-house; one of such defect tracking system is as Bugzilla1, which have become very popular now days, main reason for the popularity of the Bugzilla is because of the spread of open source software development. These type of defect tracking system helps in tracking the defects along with code change, submission of patch, review comments which helps in managing the quality assurance more efficiently and easy communication between different stakeholders [6].

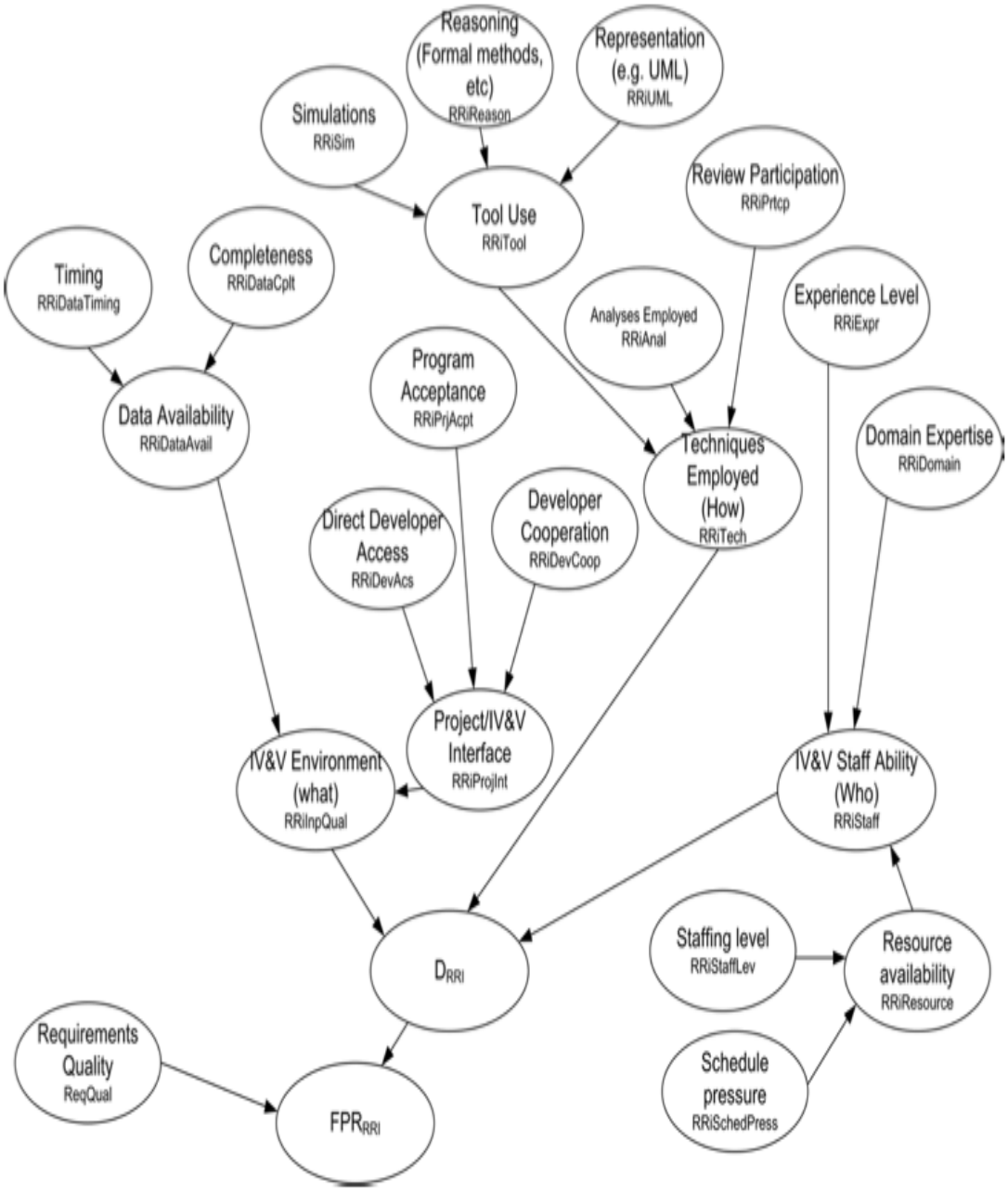

Figure 2. Defect removal

NASA usage a scale system which gives severity from 1 to 5 to assign issue severity. The scale was ranging from 1 to 5, worst to dullest [1], respectively. Table 1 represents the different scale which is used for human-rated and robotic missions.

Table 1. Severities for Robotic Missions

Defect Severity definition in Robotic Missions
Severity 1: Most critical issues, impact on safety. Some of the most critical requirements missed.
Severity 2: Failure of some critical requirement and there is no alternate or work-around. Impact on time and cost.
Severity 3: Failure of some critical requirement and there are some alternate or work-around available. Impact on time and cost.
Severity 4: UI related issues. May be inconvenience for the user. All the required functionality of the system is working fine.
Severity 5: All other issues which are not of the types mentioned above.
Defect Severity definition in human-rated missions
Severity 1: Very critical issue. Can cause impact or injury on human being.
Severity 1N: Very critical issue but occurring in the specific scenario.
Severity 2: Issue because of which critical functionality is missing.
Severity 2N: Issues with above scenario but happening in some specific scenario.
Severity 3: All other failure which are not severity 1 or 2.
Severity 4: Any other issues which cannot be easily detectable and of not the above severity levels.
Severity 5: Not any issues but required to be corrected for the future use.

The study presented in these papers for using SEVERIS system is to assess the defect severity of reported issues. These data were collected from five NASA robotic missions system [2]. The major thing which got concluded from this case study was that for generating

the severity assessment the better candidate can be unstructured text rather than the base available in structured data base.

The paper presents the major finding of this work was the efficiency and success of the stemming solution and its components. It is the combination of rule learning methods and standard text mining solutions. Because of its simplicity it makes it easy to use and can be easily adapted to other data sets. Additionally it also helps to provide relatively better estimates only by analyzing training data which got collected from multiple reports.

Chapter 3

TEXT MINING

In text mining before we apply any machine learning to the results it is very important to get the number of attributes or dimensions reduced in the given problem [5]. We can use several methods for dimension reduction which are usually used in text mining like

1. Tokenization
2. Stop word removal
3. Stemming
4. Tf*idf
5. Info Gain

3.1 Tokenization:

A block of text called as token and it is considered as most useful part of any unstructured text. We can represent Token by a syllable, a sentence, a paragraph, but in case of text mining mostly token is considered as a word. Converting a stream of characters into a sequence of tokens is a process which is represented by tokenization. We can say that in the Tokenization capitals, punctuation and brackets etc. get removed.

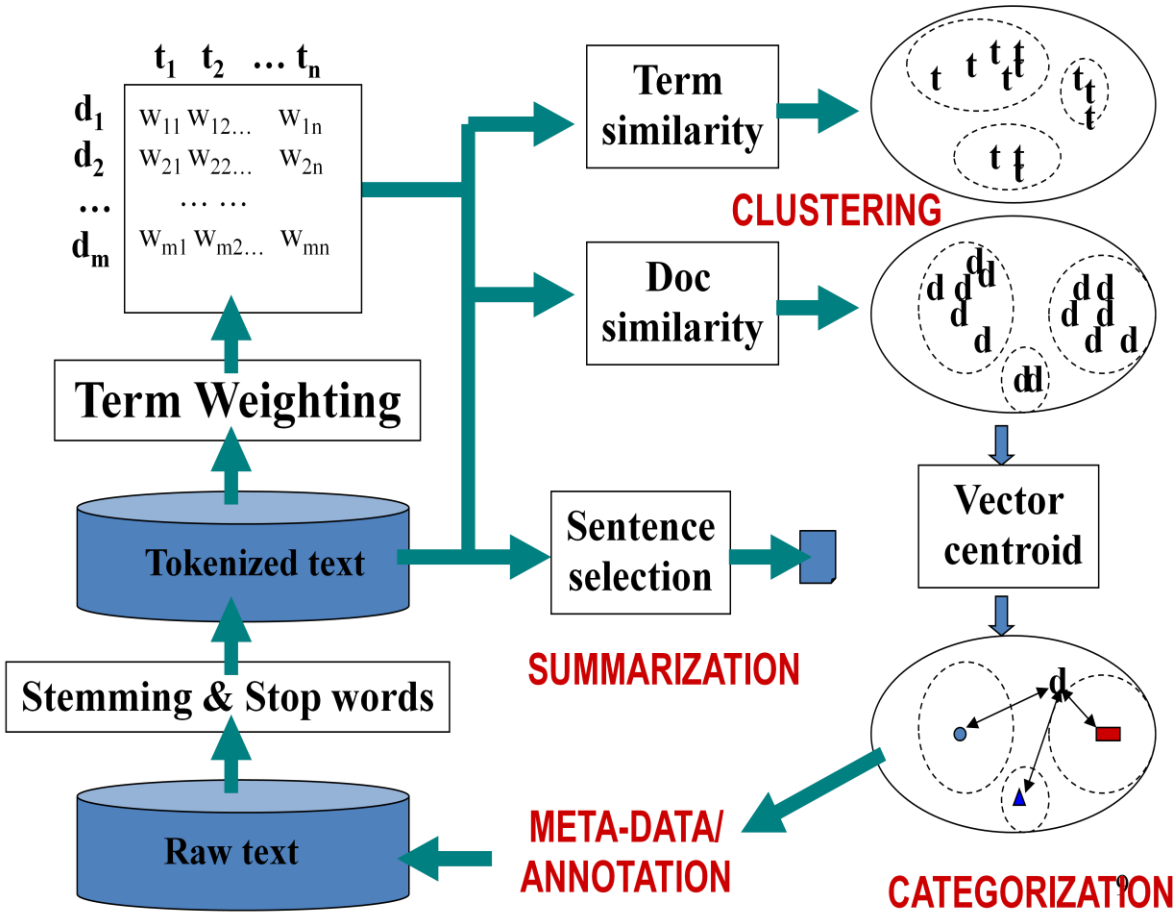

Figure 3. Generality of Basic Text Mining Technique

3.2 Stop word removal:

There are lots of commonly used words which do not have any information which is relevant to a specific context; these types of words are called as Stop Words. In the English language we can say that list of stop words are conjunctions, common verbs, pronouns, prepositions, articles, nouns, adjectives and adverbs.

a	about	across	again	against
almost	alone	along	already	also
although	always	am	among	amongst
amongst	amount	an	and	another
any	anyhow	anyone	anything	anyway
anywhere	are	around	as	at
...	...	...	...	...

Figure 4. 24 of the 262 stop words used

3.3 Stemming:

When we reduce the derived word to their base, root or stem is called the process of stemming. In this process the words written in different word form are conventionally written as same word which forms the stem. For example "eat", "eats", "ate" and "eating" can be written as "eat". The purpose of stemming is to form the same root for all words which are derived from the base word. It is not necessary that stems have to be identical words; in this case generally we map all the words to same root even if stem is not represented as a valid root.

3.4 Tf*idf:

"Term frequency times inverse document frequency" is called as "tf*idf". In case of tf*idf the weights which get assigned for a word is generally used in retrieving the information and used for text mining. To evaluate the importance of any word statistical measure of the tf*idf weights are used. General rule is based on times a word is getting used in the training data its importance keeps on increasing. In case of longer documents we need to usually normalize this count to avoid the biasness with respect to longer documents and the term with in a particular document.

Importance of any term which can get by dividing the number of all documents by the number of documents containing the term and afterwards we take the logarithm of that quotient is known as inverse document frequency.

If say that in any document the Words number = J appears Word [J]; if Doc [J] be the documents containing J, then:

Tf*idf = Word[j]/Words*log(Doc/Doc[j])

The main purpose is gathering all the words which are very important terms i.e. occurrence of these words are quite high, and we can ignore the rest of words for analysis from which we cannot get infer information.

3.5 InfoGain:

After the tf*idf InfoGain is applied to training data. In case of InfoGain measure we can say that the best words. These words which ease the target. We can also use statistical approached to determine the concept more clearly.

For example if we have a training data set which have severity=5 issues around 70% of the total issues and severity=1 issues around 30% of the total issues reported, then in this case the training data set denoted with class c (1) = severity5 & c (2) = severity1 and we can assign the frequencies as n (1) = 0.7 and n (2) = 0.3.

Where we get highest information then its attribute is highest gained information, it is given as:

$$InfoGain(A_i) = H(C) - H(C|A_i)$$

A significant dimensionality reduction is achieved by InfoGain; in this case all the text gets ranked in the training set.

Recall and precision are inversely related. If we want to increase one then we have to pay the cost of reducing the other. It is totally depends on any application, precision may be favored over recall, or vice versa.

Chapter 4

PROPOSED FRAMEWORK FOR DEFECT SEVERITY PREDICTION

It is important to minimize the no of words in the training data set to get the classification properly done. There are lots of text mining techniques available which can be used to reduce the unnecessary words or we can say that words which are not helpful in the classification can be removed. There are lots of common words which are useless for classification can be removed and it is called as stop word removal. Before creating bag of words all occurrence of these types of words can be removed by stop word removal.

There are lots of commonly used words which do not have any information which is relevant to a specific context; these types of words are called as Stop Words. In the English language we can say that list of stop words are conjunctions, common verbs, pronouns, prepositions, articles, nouns, adjectives and adverbs.

Even with the Stop Word removal it is not possible to get the required set of word which can be used for classification so reduce it further can be achieved by Stemming. Number of words present is document get analyzed in case of Stemming. The purpose of stemming to find out the set of words which can be treated as similar or equivalent words. Like we can say 'applied', 'applying', 'applies' and 'apply' are similar words.

When we reduce the derived word to their base, root or stem is called the process of stemming. In this process the words written in different word form are conventionally written as same word which forms the stem. For example "eat", "eats", "ate" and "eating" can be written as "eat". The purpose of stemming is to form the same root for all words which are derived from the base word. It is not necessary that stems have to be identical words; in this case generally we map all the words to same root even if stem is not represented as a valid root.

It is usually happens that even after removing the unnecessary words using stop word and doing stemming for rest of the texts available in the training data then also the number of words available are large making the classification difficult. In this case we can use confusion matrix. Confusion matrix is shown in the Table -2. By using confusion matrix we can reduce the number of the words which are getting used for a particular category. Using confusion matrix information gain can be calculated for the words based on its occurrence in the given category. Based on the occurrence frequency weights can be decided. Using this we can find out the words from which we can infer most useful information.

For each type of data category C_k confusion matrix can be constructed as below table.

Table 2. Confusion Matrix

		Predicted class	
		C_k	Not C_k
Actual Class	C_k	A	C
	Not C_k	B	D

After the confusion matrix gets generated for each of the all the category items it get combined to get average. Based on confusion matrices elements Recall and Precision, can be computed.

Chapter 5

EMPIRICAL STUDY

In case of Text Classification we may have N categories which may belongs to different fields may be it is from Business or from Medicine or from Management or any other filed but their training data can be part of N categories. Collection of training documents can be constructed in 2 possible ways where multiple classifications get used. These two approaches are:

- Local Dictionary Approach
- Global Dictionary Approach

Local dictionary get created for each category separately whereas global dictionary includes all the words which are appearing in any of the category. Global dictionary can be constructed faster than the local dictionary.

The severity prediction problem studied in this paper is defined as a classification problem on incoming defect reports with a classification function f_j to predict the severity level

s_i of each incoming defect report r_x:

$$s_i = f_j(r_x)$$

The prediction function f_j can be constructed using different machine learning algorithms.

There are two categories of indicators extracted in the feature selection schemes: severe indicators and non-severe indicators. The severe indicators are defined as the representative

words extracted from the historical severe defect reports with high term frequencies. Similarly, non-severe indicators are extracted from the historical non-severe defect reports.

These indicators are expected to have high discriminability for the prediction work.

To evaluate the effectiveness of the feature selection schemes, we use two major severity categories in this study to investigate the effectiveness of feature selection schemes as discussed in [2], [3]. The set S of the severity categories is thus $\{s_0, s_1\}$, where s0 denotes the non-severe class, and s_1 is the severe class. Since we also take the same datasets maintained with Bugzilla for the experiments, s_0 is currently ranged from the Bugzilla severity level trivial to minor, and s_1 is ranged from major, critical, to blocker. For performance evaluation, we adopt the 10-fold cross-validation approach to assess the prediction performance. In the experiments, however, the defect reports labeled normal were discarded as discussed in [2] because the normal level is the default option and thus many defect reports labeled do not have the correct decisions on the severity levels according to their manual observations. Considering the defect reports labeled normal in the experiments needs many human efforts to validate the perdition results. Thus the investigation of the normal-level defect reports is left to our future study.

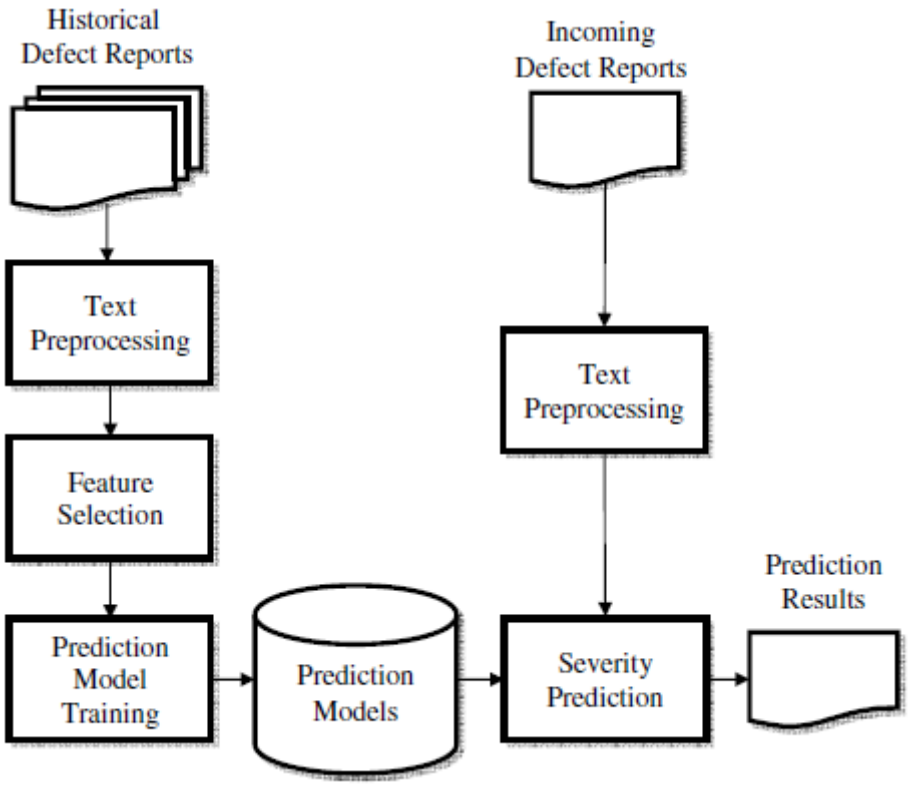

Figure 5. Process flow of Severity Prediction for incoming Defect Report

Chapter 6

DATA COLLECTION & EXPERIMENT DESIGN

Experiment Design was conducted on the real testing data being used in one organization. There are multiple fields available in the testing database and from these fields following fields taken for evaluation: issue tracking id, registered date, progress status title, severity, occurrence frequency, test place, software version, problem description, number of rejections, resolve date, resolve in charge, cause and countermeasure.

Different set of data merged together based on the severity and with particular severity the words collated and used for further analysis. To simply this a tool was made and text mining basic principle applied to that tool. At each stage the output get collated and used as input at the next level.

6.1 Objects of Experiment Design

This analysis was applied to 2 different types of data collected from different sources. In one case NASA data from the PITS C is used and in this case the analysis is done for top 5 data set, top 10 data set, 25 data set, 50 data set and top 100 data sets. This data set has been collected for 323 different projects. Other data set is taken from the one organization working in the mobile domain. The analysis got done on the top 100 data set which got collected from the 149 different projects. There data set were input to tool made for doing standard text mining techniques and final result of this tool is used as input for SPSS (Statistical Package

for Social Science) tool from IBM. This tool is used to apply machine learning techniques on the training data set. The input for SPSS is in the form of XLS file.

6.2 Tool for standard Text Mining Techniques

This tool got developed in VC++ and used for mainly applying the standard text mining techniques on the test data set. This tool takes text file as initial input files either single file or multiple files can be inputted. This tool then remove the stop words, do the stemming, tf*idf, applying the InfoGain and finally normalizing the data set. The output of this tool can be easily copied to XLS file which can be used for applying the machine learning techniques on the training data set. Screen shot for this tool is given over here to further describe this tool.

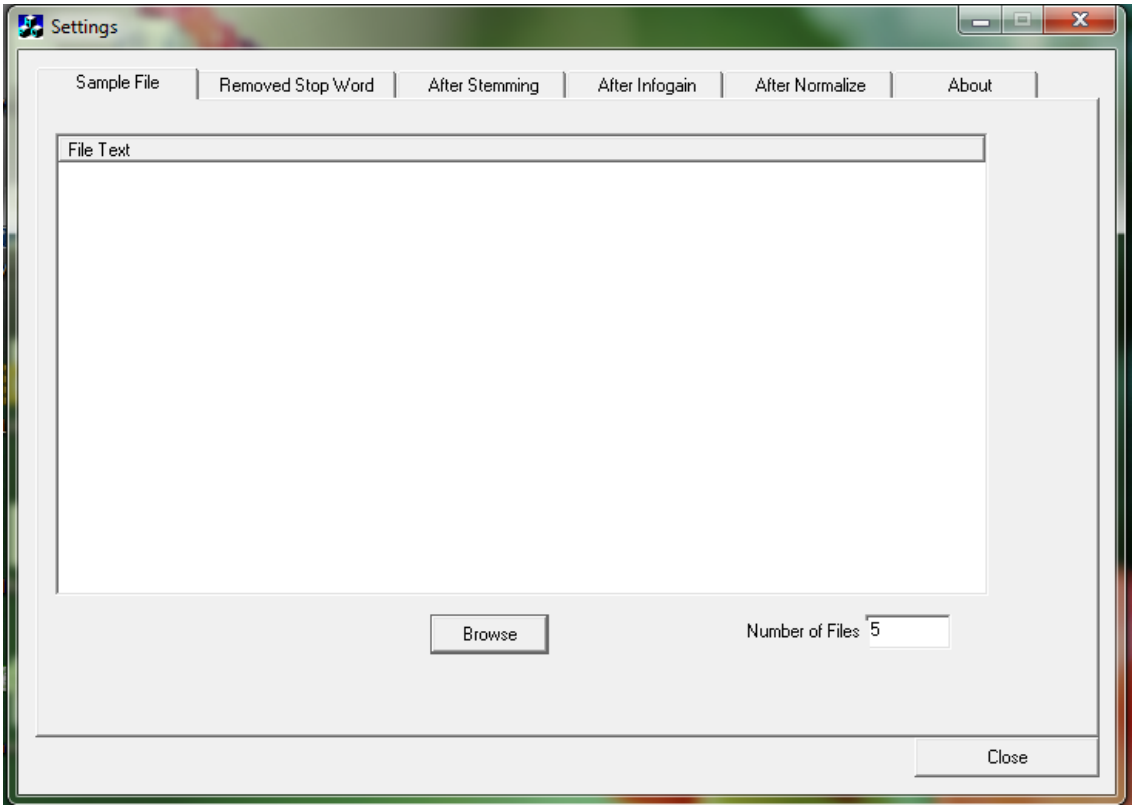

Figure 6. Initial Screen

On the initial screen user can select the multiple files in the text format can be selected. User needs to input the number of files then files need to be browsed. The selected files will be uploaded to the tool. These data are getting used in the text mining techniques.

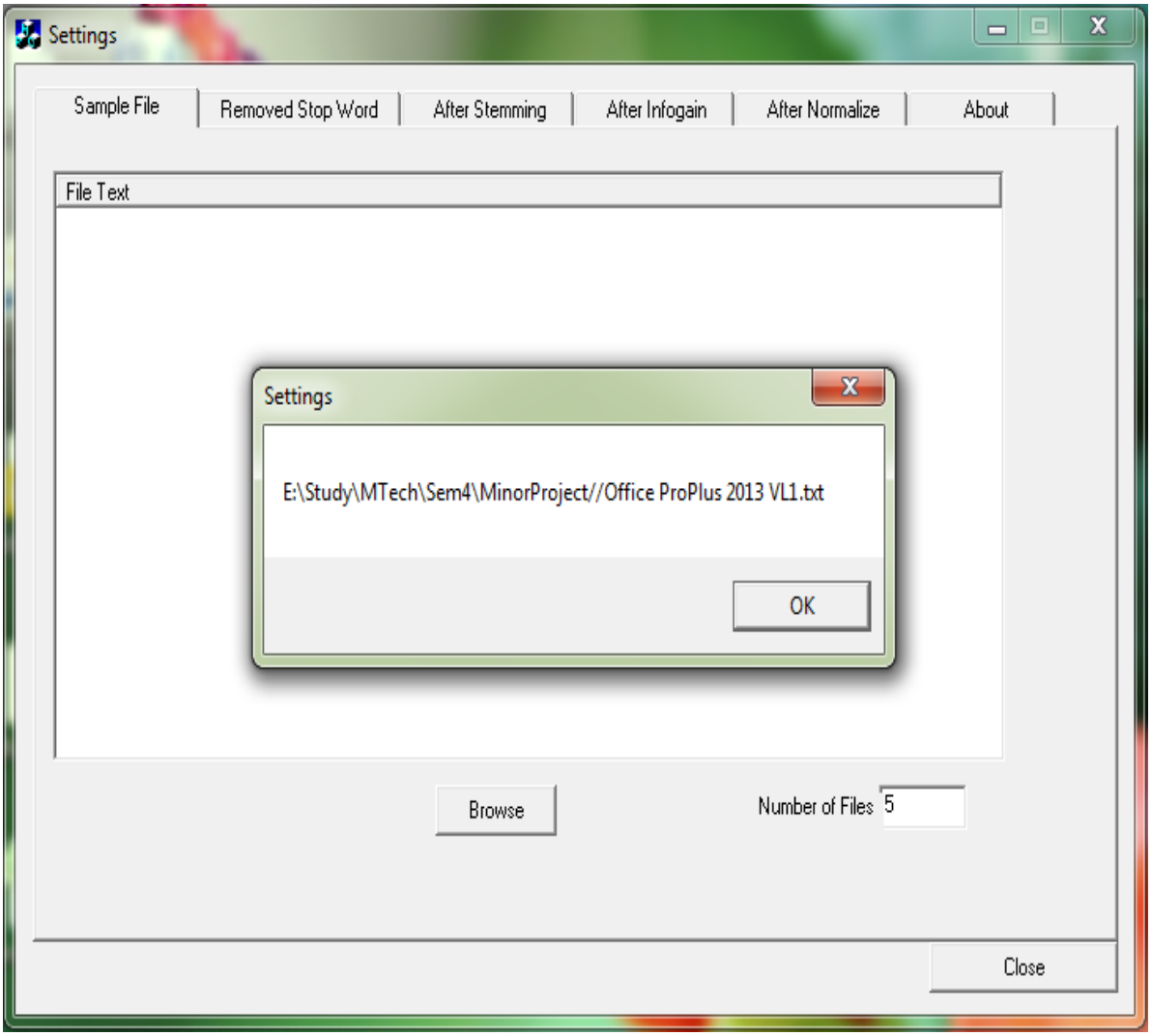

Figure 7. After the file has been selected

Figure 7 is representing the screen after the file get selected asking for the final confirmation. After getting the conformation from the user the file get uploaded for the further processing.

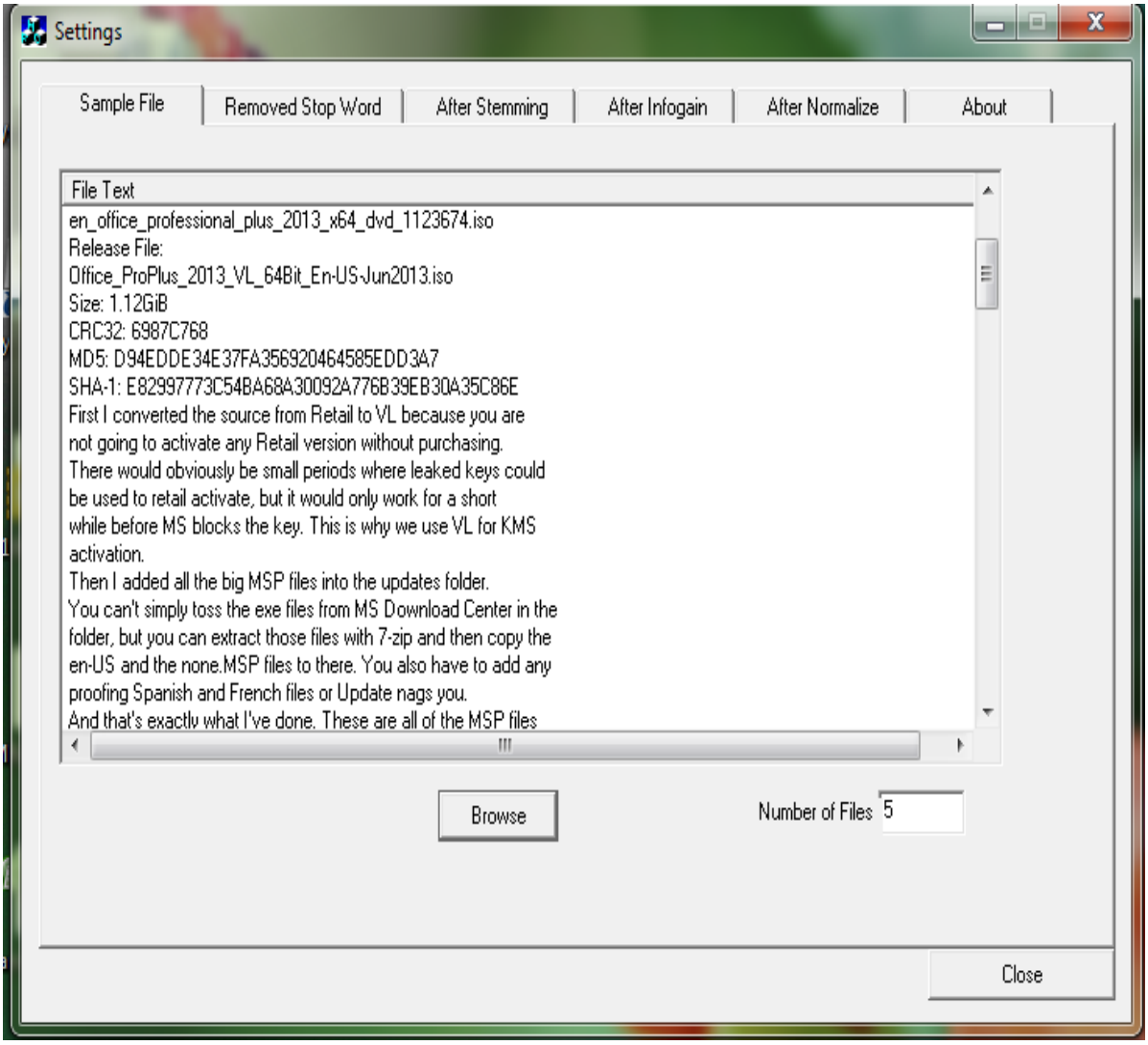

Figure 8. After the File has been loaded

Figure 8 displays the screen shot after the file has been loaded on the screen. The data from the text file get copied and displayed.

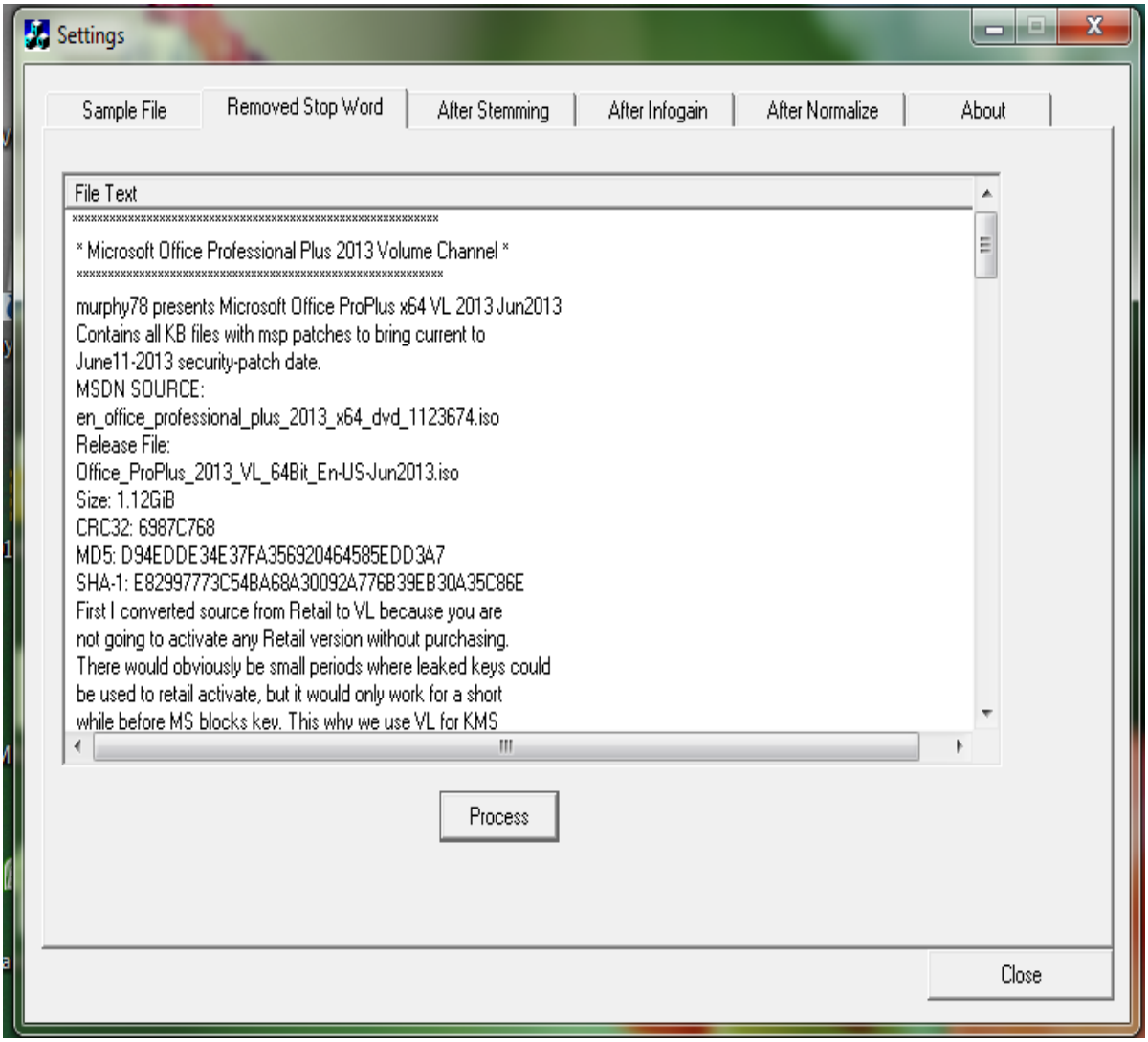

Figure 9. After Stop word removal

After the files get selected when we apply stop word then the common word gets removed from the screen. Figure 9 shows the screen after the stop word removal.

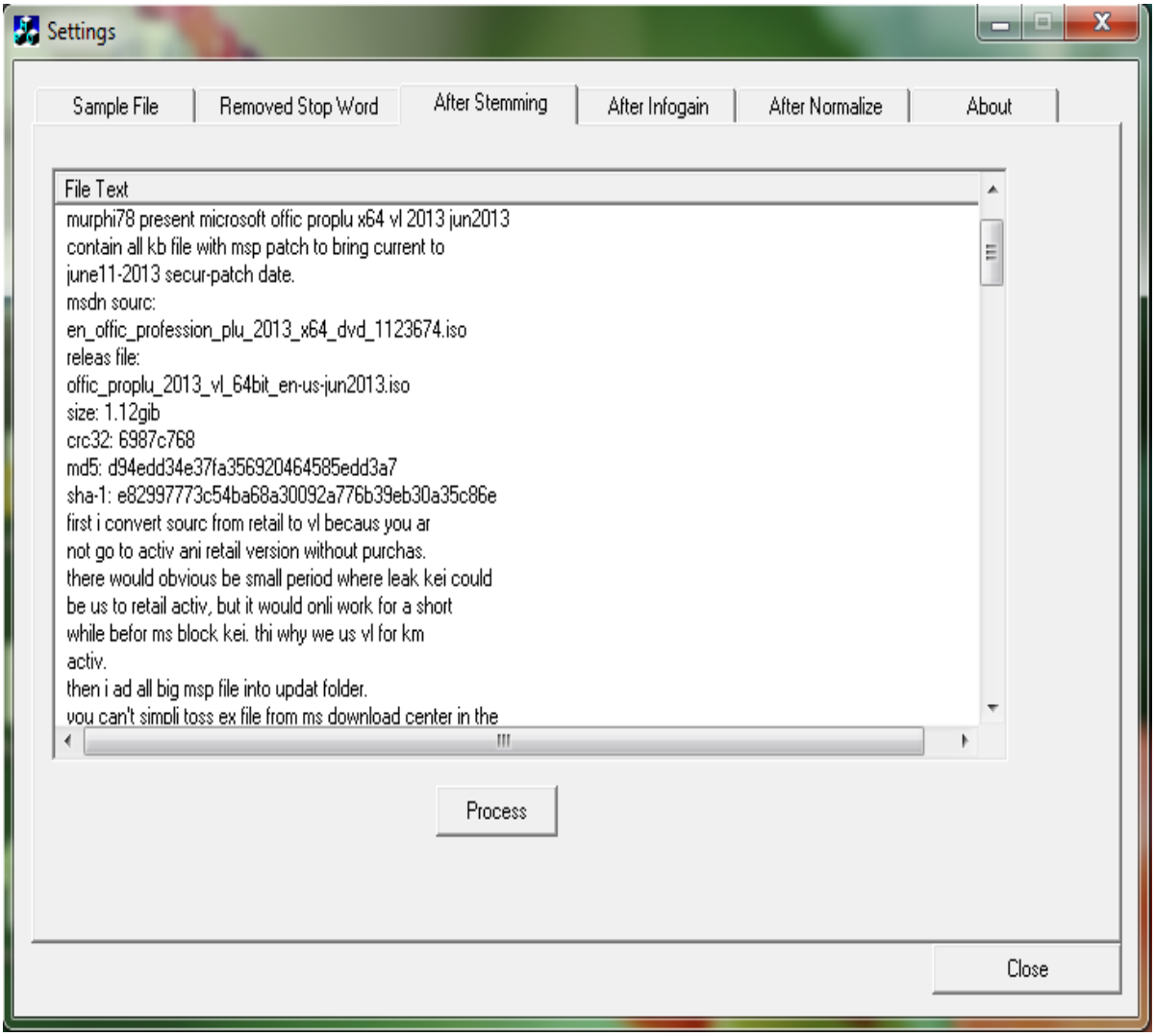

Figure 10. After Stemming

After the stop word removal, stemming is applied. In case of stemming the related word get grouped together. Figure 10 is showing the screen after stemming is applied.

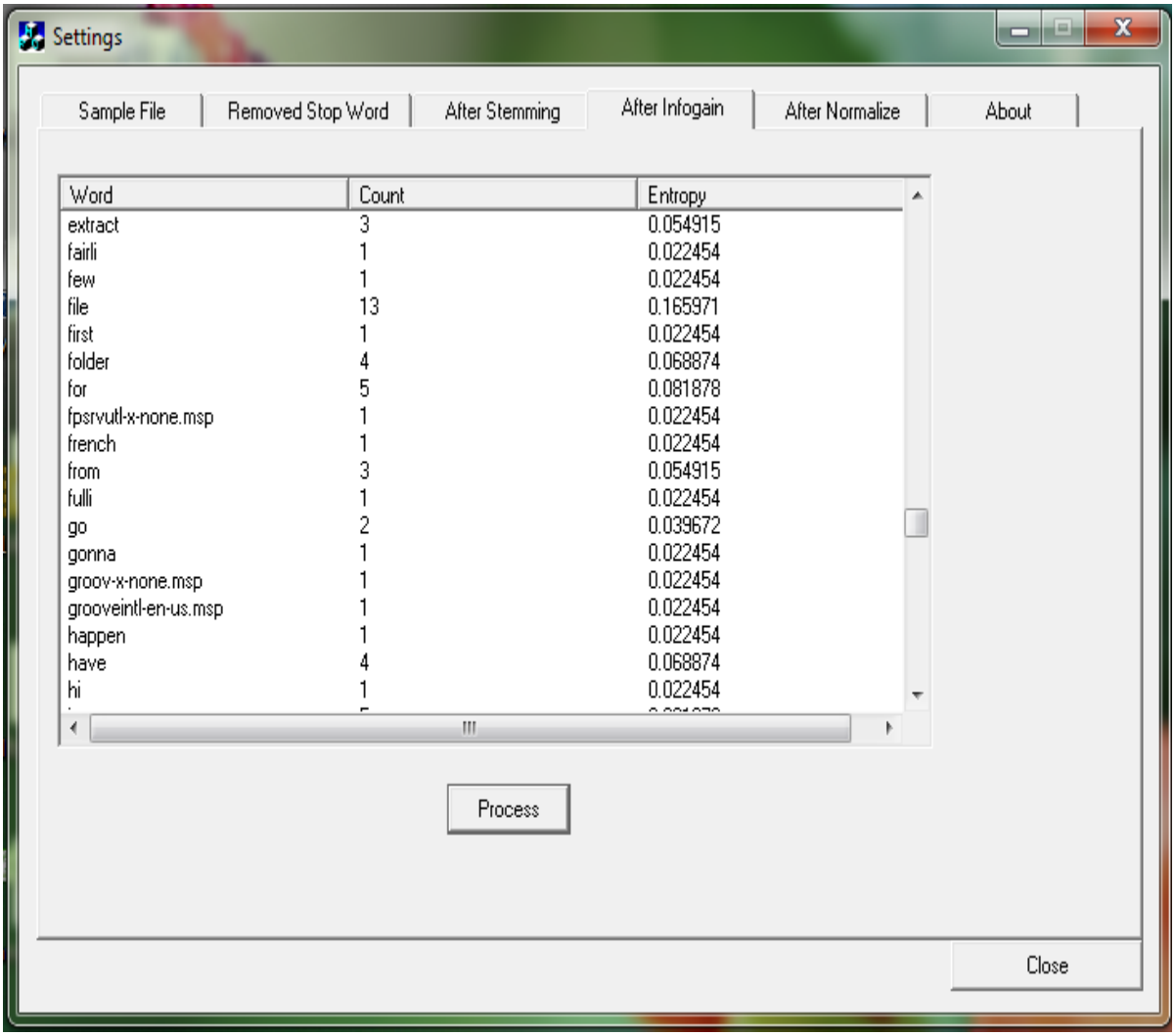

Figure 11. After applying InfoGain

After the stemming the InfoGain is applied to the data. In the InfoGain the weightage is applied to word based on the frequency of the occurrence of the words. Figure 11 is displaying the screen after applying the InfoGain.

6.3 Tool for Machine Learning Techniques

For the further processing on training data set the SPSS (Statistical Package for Social Science) tool from IBM [20] is used. This tool is mainly used for researcher for analysis of data. This tool is very popular in many fields; health science and marketing are the areas where this tool is getting used extensively. There are two views Data View which looks like spreadsheet and Variable View which is used for Metadata View. This too can read the data either from the text file, spreadsheet or from the database. Output can be given in the form of PDF, HTML or graphic image format.

In this case data is input in the form of XLS file after inputting the file the data get analyzed using classify technique (decision tree). In this case all the keywords have been taken as independent words and defect severity is taken as independent words. After the analysis the data is used for validation. Here parent node is taken as 10 and child node is taken as 5. Based on this given information the probabilities get predicted.

After the analysis of the data ROC curve, AUC values, sensitivity and 1-spsecificity has been generated this gives the prediction results.

Chapter 7

RESULTS ANALYSIS

Test dataset have been taken from 2 different sources. First data is used from one organization working in mobile domain field. The severities used in this organization are from 1 to 3 where 1 is most critical and 3 are least critical. The bug tracking tool which is used in this organization is in-house developed tool. Second dataset is used from NASA test database and PITS C data is used.

Self made tool is used to execute initial text mining techniques then the output is used in the IBM SPSS tool [20] for the machine learning and further analysis. In this tool Classify (Decision Tree) has been used for the analysis of data. Results are based on the ROC curve, AUC values, sensitivity and 1-spsecificity.

In first dataset top5, top 10, top 20, top 30 and top 50 words used which are collected from 149 different projects. The result for this dataset gives the prediction for severity 1, severity 2 and severity 3. In the second case where the PITS C dataset is used top 100 words has been taken for the analysis from 323 different projects. Analysis has been done on top 5 words, top 10 words, top 25 words, top 50 words, top 75 word and top 100 words.

7.1 Test data 1

This data is collected from one leading organization working in the mobile domain. The data set is having 50 top keypads and data go collected for 149 projects. After getting the data reduced using the standard text mining techniques the data is input to SPSS tool.

7.1.1 Result Summary

Table 3. Result Summary with Top 5 Dataset for Test Data 1

	AUC	Sensitivity	1-Specificity
Sev 1	0.660	1	0.225
Sev 2	0.609	1	0.178
Sev 3	0.645	0.867	0.576

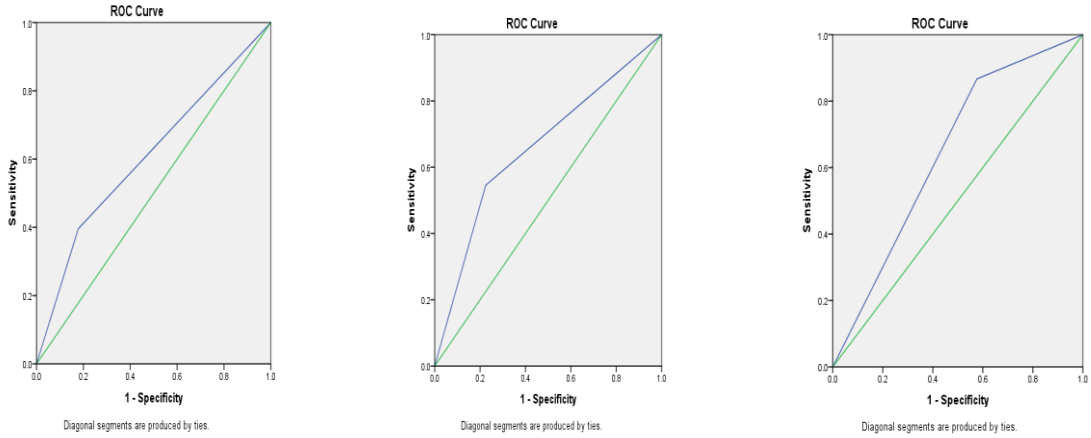

Figure 12. ROC Curve corresponding to Top 5 Dataset – Test Data 1

Above result shows, AUC needs to be improved. This is possible by checking algorithm for Stemming and confirming correctness of Info Gain calculation. All the AUC values which have been analyzed for all severities level indicates that may be the dataset is very less because of which the prediction is not coming good. Top 5 words have been chosen from the test dataset 1 for the analysis.

For test dataset, result on running IBM SPSS tool mentioned here for reference:

For dataset 1, result on running IBM SPSS tool for dataset 1 produced from Text Mining Tool.

Table 4. Result Summary with Top 10 Dataset for Test Data 1

	AUC	Sensitivity	1-Specificity
Sev 1	0.812	0.818	0.348
Sev 2	0.675	0.792	0.515
Sev 3	0.691	0.967	0.220

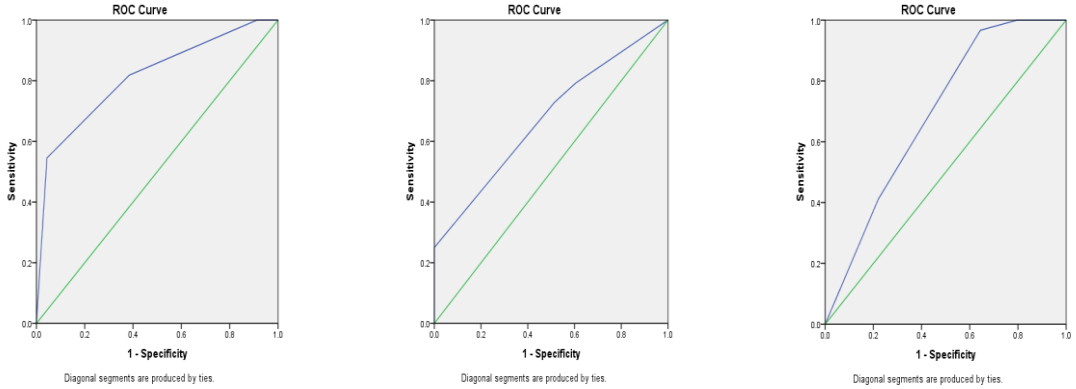

Figure 13. ROC Curve corresponding to Top 10 Dataset – Test Data 1

In this case AUG for Severity is 0.8 which shows the good prediction but the severity 2 AUC value need to be improved. With the slight improvement the severity 3 also predicts the good result. These data are presented in the ROC curve shown in figure 13.

Table 5. Result Summary with Top 20 Dataset for Test Data 1

	AUC	Sensitivity	1-Specificity
Sev 1	0.792	0.727	0.268
Sev 2	0.653	0.479	0.218
Sev 3	0.701	0.967	0.475

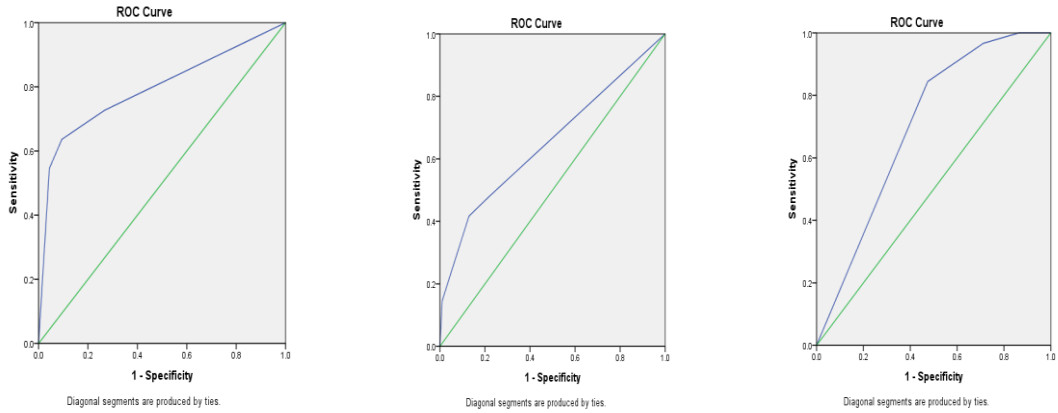

Figure 14. ROC Curve corresponding to Top 20 Dataset – Test Data 1

Above result shows, AUC needs to be improved for severity whereas it is fine for severity 1 and 3. Severity 1 prediction results are good. These data are presented in the ROC curve shown in figure 14. This result is better than the above 2 results. One main reason is in this case top 20 words have been taken for the analysis which is resulting in the better results as compare to other 2 above results.

Severity 2 AUC values still need some more data so that the prediction results can become better.

Table 6. Result Summary with Top 30 Dataset for Test Data 1

	AUC	Sensitivity	1-Specificity
Sev 1	0.810	0.818	0.391
Sev 2	0.679	0.792	0.505
Sev 3	0.694	0.967	0.220

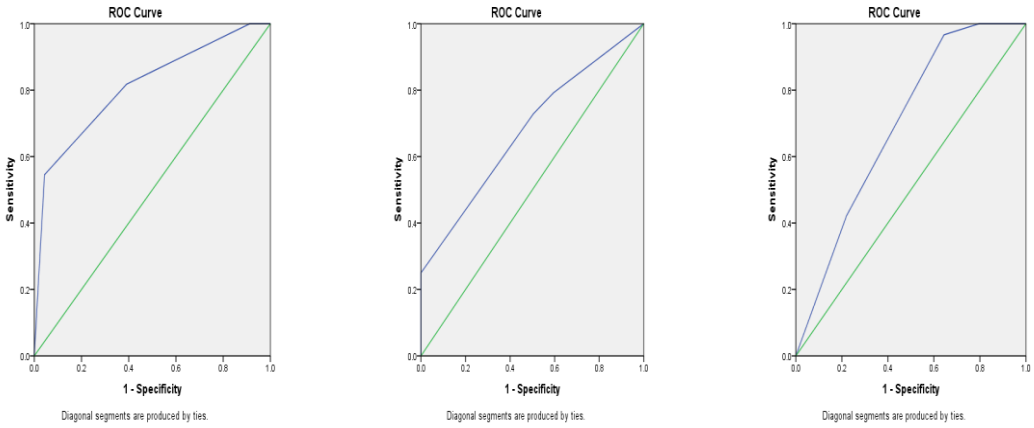

Figure 15. ROC Curve corresponding to Top 30 Dataset – Test Data 1

Above result shows, AUC needs to be improved for the severity 2 & severity 3 whereas the results for Severity 1 are good. It shows that the prediction for the severity 1 is good. These data are presented in the ROC curve shown in figure 15.

In this case top 30 words have been taken for the analysis from the test dataset 1. Because of increase in the test data set prediction for severity 1 is coming better where as for the prediction of severity 2 & severity 3 still more data is required for have better prediction results.

Table 7. Result Summary with Top 50 Dataset for Test Data 1

	AUC	Sensitivity	1-Specificity
Sev 1	0.801	0.909	0.043
Sev 2	0.631	0.917	0.119
Sev 3	0.658	0.967	0.102

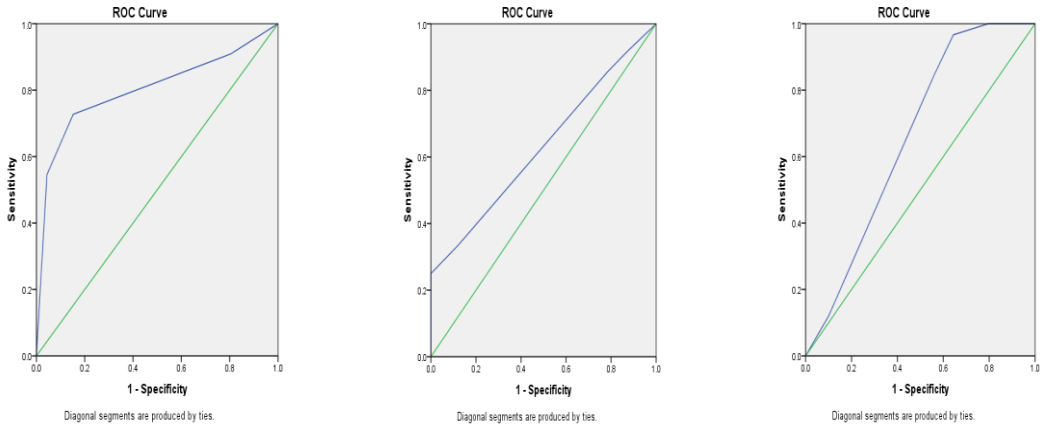

Figure 16. ROC Curve corresponding to Top 50 Dataset – Test Data 1

Above result shows, that the sample dataset with top 50 word the result for AUC for Severity 1 prediction is good its value is 0.801. AUC needs to be improved for the severity 2 and severity 3 predictions.

These data are presented in the ROC curve shown in figure 16.

7.2 Test Data 2 (PITS C)

Test data 2 is taken from NASA's PITS database. For the analysis purposes PITS C data has been taken and in this case top 100 words have been taken for 323 different projects. Severity 3, 4 and 5 used for the analysis with dataset of top 5 words, top 10 words, top 25 words, top 50 words and top 100 words.

7.2.1 Result Summary

Table 8. Result Summary with Top 5 Dataset for Test Data 2

	AUC	Sensitivity	1-Specificity
Sev 3	0.901	0.887	0.226
Sev 4	0.871	0.994	0.394
Sev 5	0.715	0.889	0.49

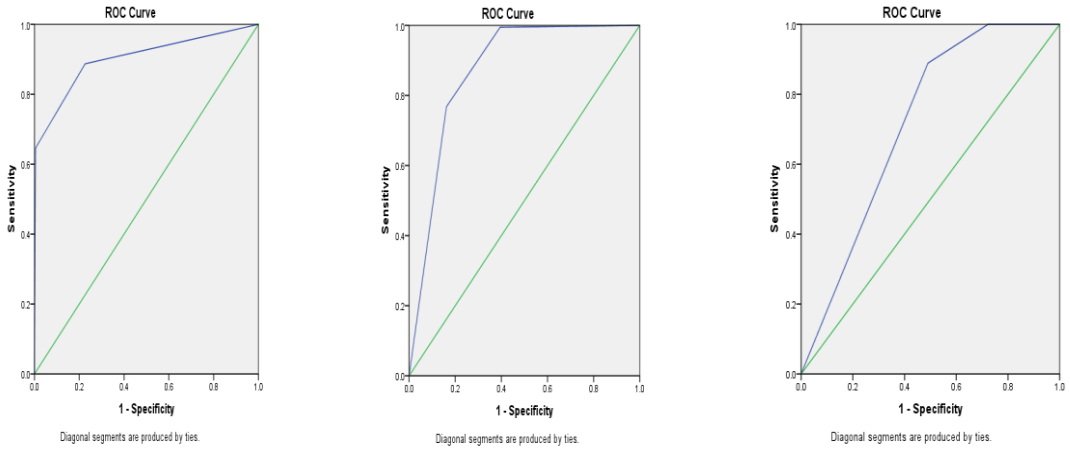

Figure 17. ROC Curve Corresponding to Top 5 Dataset – Test Data 2 (PITS C)

In the above result AUC value for Severity 3 is 0.901 which indicates that the prediction for sever1ty 3 is very good. In this AUC value for Severity 4 and 5 is also good which predicts that the prediction is good. Sensitivity with Severity 3 is 0.887. The overall results of severity 3 prediction are very good.

In this case top 5 words from the PITS C database has been taken for the analysis. ROC curves have been shown in figure 17 for all the severities levels.

Table 9. Result Summary with Top 10 Dataset for Test Data 2

	AUC	Sensitivity	1-Specificity
Sev 3	0.929	0.977	0.632
Sev 4	0.922	0.956	0.387
Sev 5	0.826	0.778	0.385

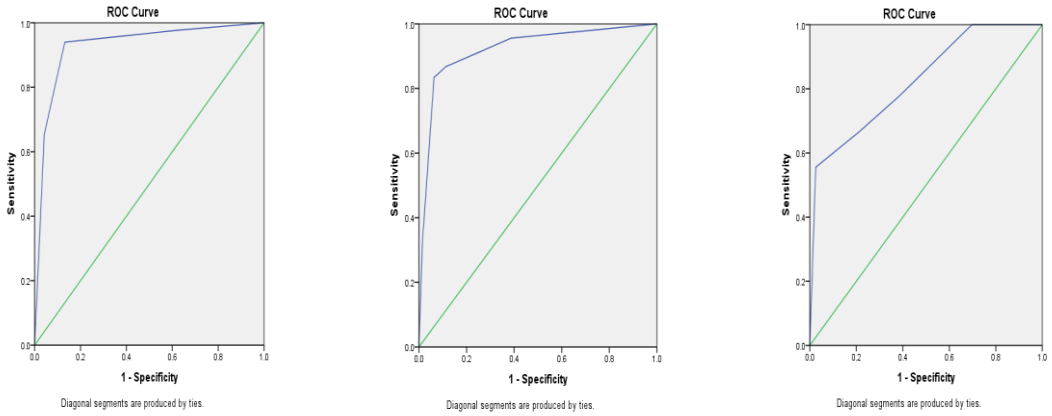

Figure 18. ROC Curve Corresponding to Top 10 Dataset – Test Data 2 (PITS C)

In the above result where top 10 dataset put under the analysis for PIS C database, the AUC results for Severity 3 and 4 is very good which is more than 0.9 and it indicates that the prediction is good for severity 3 and severity 4. The result for severity 5 predictions is also good but comparison gives that prediction for severity 3 and 4 is better than severity 5 predictions.

In this case top 10 words have been taken from the PITS C data base for the analysis. Results are better than the previous results where only top 5 words have been taken for the analysis.

Table 10. Result Summary with Top 25 Dataset for Test Data 2

	AUC	Sensitivity	1-Specificity
Sev 3	0.954	0.970	0.405
Sev 4	0.947	0.989	0.345
Sev 5	0.873	0.111	0.035

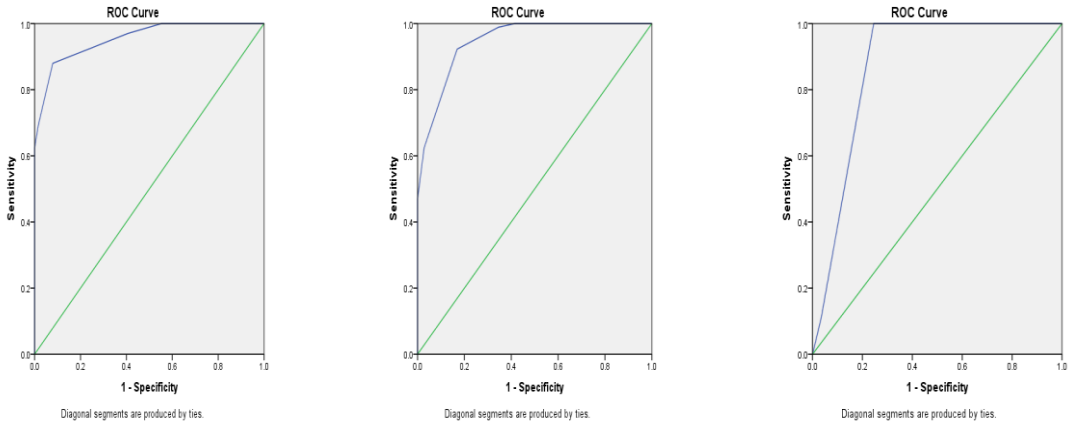

Figure 19. ROC Curve Corresponding to Top 25 Dataset – Test Data 2 (PITS C)

Above result shows, AUC for Severity 3 and severity 4 predictions is good as compare to severity 5 predictions. The moment the number of words and iteration keep on increasing the results are coming much better. Sensitivity value for severity 3 and severity is also closed to 1. In this case top 25 words from the PITSC database have been taken for the analysis. ROC curve for all the severities have been shown in figure 19. AUC results are getting better if the number of test dataset is increasing.

Table 11. Result Summary with Top 50 Dataset for Test Data 2

	AUC	Sensitivity	1-Specificity
Sev 3	0.951	0.910	0.137
Sev 4	0.941	0.989	0.345
Sev 5	0.839	0.889	0.210

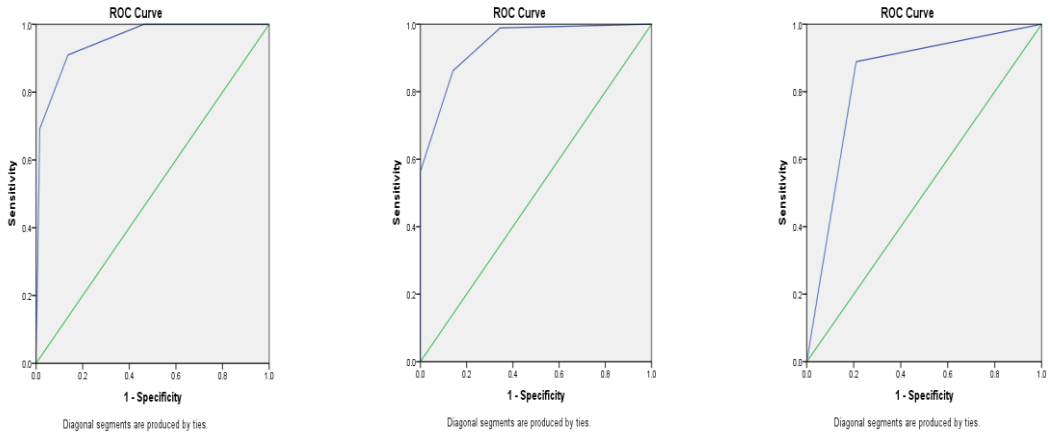

Figure 20. ROC Curve Corresponding to Top 50 Dataset – Test Data 2 (PITS C)

In the above result where top 50 dataset put under the analysis for PIS C database, the AUC results for Severity 3 and 4 is very good which is more than 0.9 and it indicates that the prediction is good for severity 3 and severity 4.

In this case 50 top words been taken in the test dataset. We can also observe that the increasing the no in the test dataset the results are getting improved.

Table 12. Result Summary with Top 75 Dataset for Test Data 2

	AUC	Sensitivity	1-Specificity
Sev 3	0.973	0.970	0.100
Sev 4	0.976	0.989	0.345
Sev 5	0.921	0.889	0.223

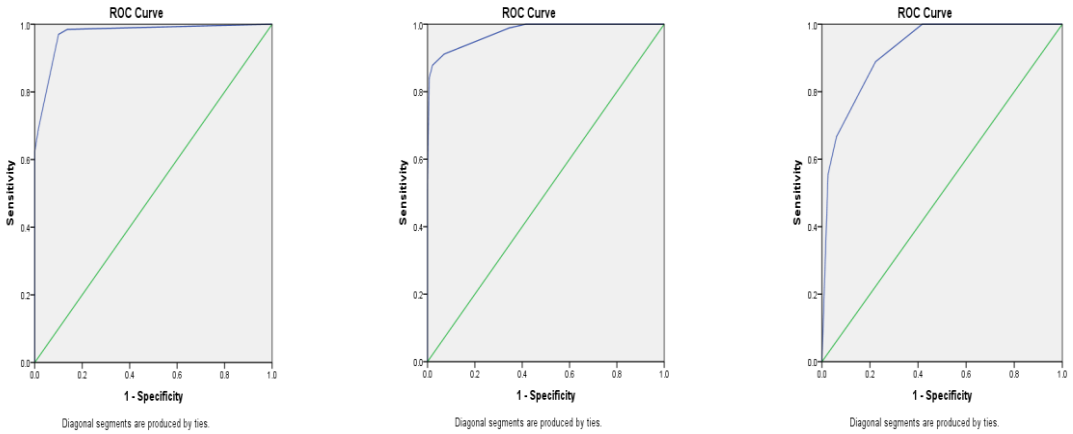

Figure 21. ROC Curve Corresponding to Top 75 Dataset – Test Data 2 (PITS C)

In the above result which was carried out on top 75 dataset for PITSC the AUC results are more than 0.9 which indicates that the prediction for severity 3, 4 &5 is good. If we compare the result for previous results then we can conclude that the prediction results are coming better if the number of words keep on increasing in the dataset.

In this case 75 top words been taken in the test dataset. We can also observe that the increasing the no in the test dataset the results are getting improved.

Table 13. Result Summary with Top 100 Dataset for Test Data 2

	AUC	Sensitivity	1-Specificity
Sev 3	0.977	0.985	0.158
Sev 4	0.976	0.989	0.345
Sev 5	0.921	0.889	0.223

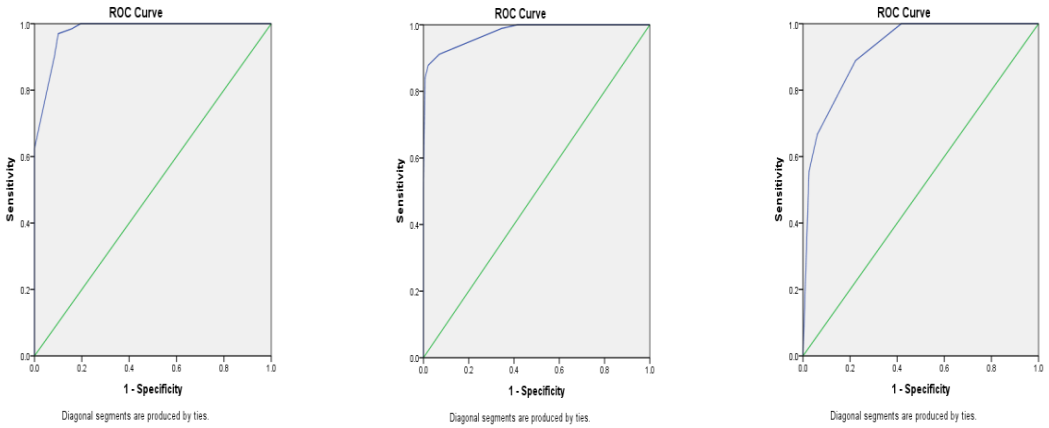

Figure 22. ROC Curve Corresponding to Top 100 Dataset – Test Data 2 (PITS C)

The above results are coming on analysing 100 top words in the PITS C test database. AUC values which are coming more than 0.9 for all the severity prediction indicates that the prediction is good. These results have been analyzed using IBM SPSS tool. When we are using top 100 words then the results and predication are best as compare to other results where lesser number of top words is used for analysis.

Chapter 6

CONCLUSION & FUTURE WORK

The Project DEFECT SEVERITY PREDICTION USING TEXT MINING is used for predicting the severity of the defects automatically. It will help in clearly defining the severity of the defect and will be consistent. For the analysis purposes the training data have been taken from two different sources. One data is collected from one of the leading mobile software development company. Data for 149 different projects collated and analyzed for the top 100 words. Second data has been collected from NASA's PITS database. PITS C data is used for analysis. Data has been collected from 323 different projects for top 100 words. There are 2 different tools have been used for applying text mining techniques and machine learning techniques.

Tool for applying the text mining techniques is self made and after this analysis top 100 words chosen in case of the PITS C dataset and top 50 words taken from the other test data set. To analyze it further and applying machine learning IBM SPSS tool is used. Decision tree (classify) machine learning technique is used for the prediction of severity.

For different data set ROC curve have been generated along with the AUC, sensitivity and 1-specificity values. AUC values give the confidence for the predicted severities. The case where AUC value is more than 0.7 we can say that the prediction is good.

It was also noticed that the results are coming better if the number of elements get increased in the test dataset.

We can continue this study using different machine learning techniques and analyze the results further. In this case 2 dataset have been taken from different sources. This analysis can be further studies on different types of test datasets taken from multiple sources. Based on the analysis a full proof system can be made which will help software industries in predicting the severity of defects more accurately and more efficiently.

REFERENCES

[1] Tim Menzies and Andrian Marcus - Automated Severity Assessment of Software Defect Reports

[2] Tim Menzies - Improving IV&V Techniques Through the Analysis of Project Anomalies: LINKER - preliminary report

[3] Ananiadou, S. and McNaught, J. (Editors) (2006). *Text Mining for Biology and Biomedicine*. Artech House Books. ISBN 978-1-58053-984-5

[4] Bilisoly, R. (2008). *Practical Text Mining with Perl*. New York: John Wiley & Sons. ISBN 978-0-470-17643-6

[5] Feldman, R., and Sanger, J. (2006). *The Text Mining Handbook*. New York: Cambridge University Press. ISBN 978-0-521-83657-9

[6] Canfora, G. and Cerulo, L., "How Software Repositories can Help in Resolving a New Change Request", in Proceedings Workshop on Empirical Studies in Reverse Engineering, 2005, pp

[7] Porter, M., "An Algorithm for Suffix Stripping", Program, 14, 3, July 1980, pp. 130-137.

[8] Wang, X., Zhang, L., Xie, T., Anvik, J., and Sun, J., "An Approach to Detecting Duplicate Bug Reports using Natural Language and Execution Information", in Proceedings 30th International Conference on Software Engineering (ICSE'08), Leipzig, Germany, 10 - 18 May 2008

[9] Cubranic, D. and Murphy, G. C., "Automatic Bug Triage Using Text Categorization", in Proceedings 6th International Conference on Software Engineering & Knowledge Engineering (SEKE'04), 2004, pp. 92–97.

[10] Canfora, G. and Cerulo, L., "Impact Analysis by Mining Software and Change Request Repositories", in Proceedings 11th IEEE International Symposium on Software Metrics (METRICS'05), September 19-22 2005, pp. 20-29.

[11] Korel, B., Tahat, L., Harman, M. "Test prioritization using system models" in Proceedings of the 21st IEEE International Conference on Software Maintenance (ICSM 2005), 2005, p 559–568.

[12] Korel, B., Koutsogiannakis, G., Tahat, L.H. "Model-based test prioritization heuristic methods and their evaluation" in Proceedings of the 3rd international workshop on Advances in Model-based Testing (A-MOST 2007), ACM Press, 2007, p 34–43.

[13] Korel, B., Koutsogiannakis, G., Tahat, L.. "Application of system models in regression test suite prioritization" in Proceedings of IEEE International Conference on Software Maintenance 2008 (ICSM 2008), IEEE Computer Society Press, 2008, p 247–256.

[14] Horgan, J.R., London, S., Mathur, A.P., Wong, W.E. "Effect of test set size and block coverage on the fault detection effectiveness" in Proceedings of the Fifth International Symposium on Software Reliability Engineering, (November 1994), 230-238.

[15] Horgan, J.R., London, S., Mathur, A.P., Wong, W.E. "Effect of test set minimization on the fault detection effectiveness" in Proceedings of the 17th International Conference on Software Engineering, (April 1995), 41-50.

[16] Avritzer, A., Weyuker, E.J. “The automatic generation of load test suites and the assessment of the resulting software”. IEEE Transactions on Software Engineering, 21(9), September 1995, 705-716.

[17] Agrawal, H., Horgan, J., London, S., Wong, W. “A study of effective regression in practice” in Proceedings of the Eighth International Symposium on Software Reliability Engineering, November 1997, p 230-238.

[18] Chu, C., Harrold, M., Rothermel, G., Untch, R. “Test case prioritization: An empirical study” in Proceedings of the International Conference on Software Maintenance, 1999, p 179-188.

[19] Weka official site:
http://www.cs.waikato.ac.nz/ml/weka/

[20] SPSS official site:
http://www-01.ibm.com/software/in/analytics/spss/

[21] Synapse official site:
http://synapse.apache.org/source-repository.html