Validation of Inspection Reviews over Variable Features Set Threshold

Maninder Singh
Department of Computer Science
North Dakota State University
Fargo, USA
maninder.singh@ndsu.edu

Gursimran S. Walia
Department of Computer Science
North Dakota State University
Fargo, USA
gursimran.walia@ndsu.edu

Anurag Goswami
Department of Computer Science
Bennett University
Greater Noida, India
Anurag.goswami@bennett.edu.in

***Abstract*—Background: Mining software requirement reviews involve natural language processing (NLP) to efficiently validate a true-fault as useful and false-positive as non-useful. Aim: The aim of this paper is to evaluate our proposed mining approach to automate the validation of requirement reviews generated during an inspection of NL requirements document. Method: Our approach utilized two training models; one from requirement reviews and other from online movies. We conducted an empirical study to test our approach using part of speech (POS) against these two trained models and observed trends w.r.t. F-measure and G-mean along with percentage of features used to train two models. Results: The results showed that using training reviews from two different domains report similar trend across evaluation metrics. Our results show that the most stable and promising validation results for F-measure and G-mean are obtained when a model over inspection and movies reviews are trained using feature set threshold value 65% and 45% respectively.**

Keywords—feature sets, faults, inspection reviews, machine learning, part of speech, sampling, class imbalance.

I. INTRODUCTION

To deliver high quality software, effort is spent to identify and fix faults during the early stages (e.g., requirements and design) of development [1], where they are cheapest and easiest to fix and saves significant re-work costs during the later stages (e.g., coding and testing). A commonly accepted practice of identifying early software development faults is by employing software inspections; where an inspector reads through the software artifact being inspected and log faults into the fault form [2, 27]. The faults that are reported contains some false-positives i.e. reviews that are not fault but are reported as faults. Many studies have reported inspection to be efficient up to 50%-70% making it susceptible to high false-positives [21].

To overcome this problem, many researchers have contributed with their machine learning approaches to automate usefulness of reviews [3-5]. The challenge to automate post-inspection process with classification techniques is small volume of reviews that contains imbalanced ratio between outcome classes (fault or false-positives); refers to as class imbalance problem. Many researchers address this challenge through sampling, ensemble and adopting unbiased metrics to evaluate a model [6-10]. Our *proposed solution* is to build on ensemble techniques; expand them to include part of speech (POS) tags and feature-sets in natural language (NL) text; to study any improvement at classification and comparing the aforesaid technique with the model trained on reviews from semantically similar domain of movies to validate requirement reviews.

Additionally, the *motivation to* explore part of speech (POS) tags w.r.t Natural Language (NL) context was to perform analysis over various text features that could result in better prediction [10-11]. Our *motivation* is to include only the most informative features that are discrete within each class. We analyzed evaluation metrics by varying the percentage of features included to train the model. This research explores a threshold and most general POS tags for most informative features in requirement reviews and reviews from movies domain that gives the most accurate and precise classification results. For class imbalance problems; the evaluation of classifier through precision or recall alone cannot be expressed because the classifiers evaluation is biased to the class with majority instances [14]. So, some more reliable evaluation metrics like F-measure and G-mean are used.

II. RELATED WORK

There has been extensive research on mining unstructured text/reviews. During literature search, we found a few studies that were either most relevant to our problem domain or we build our approach on their footnotes. Our approach aimed at mining requirement reviews to validate each review either into fault or non-fault category. Inspection reviews being unstructured text required a lot of pre-processing tasks like removing punctuations, stop words, removing slangs and fixing spelling mistakes if any. From literature, we found an interesting contribution by Bavota et.al [15] in which they used pattern matching, information retrieval and natural language processing (NLP) to mine unstructured data (MUD). Using MUD techniques, they mined unstructured (raw) text of bug reports, to generate automatic documentations to improve software quality. On the same guidelines, Bosu et.al. [16], Chen et.al. [17] and Agarwal et.al. [18], presented their work to categorize a textual data into useful or non-useful. Bosu [16] presented qualitative investigation of some prominent facet (e.g. reviewers experience) associated with reviews under analysis that could help categorize a review more accurately into useful or non-useful. They build their classifier using Naïve Bayes (NB), Multinomial NB and Decision Trees (DT) and applied pre-processing tasks like stemming, removing stop words using Natural Language Toolkit (NLTK). Chen [17] used NL processing and topic modelling to categorize textual reviews from Mobile App Market place into useful and non-useful. Agarval et.al, used Twitter data (again unstructured text) to classify tweets into categories based on their polarity (positives, negatives and neutral) using Tree based classification approach, NLP over part of speech (POS) tags. Another work in classification of tweets was presented by Gimbel et.al [19] in which they analyzed Twitter tweets using POS tags. They mentioned that the most prominent POS tags that are present in any tweet are Nouns (N), Adjectives (J), Adverbs

978-1-5386-3446-2/17 $31.00 © 2017 IEEE
DOI 10.1109/MLDS.2017.16

(R), punctuations, Verbs (V) and Determiners (DT). The above discussed work related to unstructured text and use of POS tags gave us insight on conducting our study based on most effective POS tag applicable to our problem domain.

Apart from the selection of appropriate classifiers, pre-processing tasks and toolkits; our data from inspection reviews comprised of class imbalance problem that were addressed by many existing studies in Literature especially for binary classification problem. The work by Mustafa et.al. [8], Nguyen et.al. [7] and Abdelwahab et.al. [20], stipulated remedies to overcome class imbalance problem. Mustafa in [8] describes the use of boosting and ensemble approach to address class imbalance. Boosting is meta-algorithm in machine learning to reduce bias and variance in supervised learning. Output from many weak learning classifiers is combined into weighted sum that represents the outcome as one strong classifier. Ensemble on the other hand use multiple learning algorithms to obtain better predictive performance. The studies in [7, 20] also emphasis that for class imbalance problem; some unbiased evaluation metrics should be used e.g. F-measure, G-mean, AUC-ROC etc. One key feature used by [7, 20] is that they considered variant training set size. The variant training set size chosen by Abdelwahab is 10 equal-sized intervals consisting 10% of the training data in each interval. Their aim was to obtain the outcome of evaluation metrics over each training set interval to analyze performance trend. The analysis of training data over variant intervals from these studies became a key research question in our study.

III. PROPOSED SOLUTION

Our proposed mining approach used supervised learning classifiers, reviews (requirements and movies) and features generated during study run to evaluate the accuracy and G-mean (Geometric mean of precision and recall). Two types of review data sets (requirement reviews and online movie reviews) are used in this study to develop two different training models (*inspection trained* vs. *movie trained* respectively). Movie reviews were included to:

- Remove the class imbalance problem in inspection trained data by introducing balanced class; and to analyze if large volume of natural language (NL) data could add valuable features required to classify requirement reviews.
- Study the impact of training classification model on reviews from semantically similar domain of movies (in context of NL text) to test requirement reviews.
- Study the minimum required feature-set threshold value for movie reviews that could produce at least similar or better results (G-mean and F-measure) when compared to training a model over imbalanced requirement reviews.

The reviews generated online for movies are 10 times larger than software requirement reviews, are more readily available online and are being evaluated for addressing class imbalance problem in mining. The reviews were divided equally into two categories: positive and negative. Positive reviews in context to movies used positive-feedback to describe a movie while negative reviews used detrimental feedback to describe a movie. We used '*positive*' category of movie reviews to train the model to detect false-positives and '*negative*' category to detect true-faults.

The features were extracted from both '*movies*' and '*inspection*' reviews. The whole feature-set for both the reviews ('*movies*' and *'inspection'*) was divided into 20 intervals (5% to 100%) of equal size to perform analysis within each feature-set interval. The model was trained 20 times by gradually increasing the size by 5% each time until all features were used to train the model. The '*movies trained*' and inspection trained model had a total of 11318 and 368 features respectively. The following example explains the idea behind extracting features with the help of NL text.

Feature set generation: Feature set generation is explained with an example (Table 1) using three reviews. Two reviews in Table 1 are true-faults while one review is a false-positive. Each word in these reviews is a feature and collection of all the features from all the reviews in each category makes feature set.

TABLE 1. EXAMPLE OF FEATURE GENERATION

Reviews	Category
The working of system in heavy load is not tested.	Fault
System load is not tested.	Fault
Who opened the gate?	False-positive

The extracted feature set is shown in Table 2. The feature "system:2" is interpreted as "system" being the unique feature and the number after colons in "system:2" denotes the frequency of that feature in a fault class. A similar feature set as generated in Table 2 is used to train the models using supervised learning classifiers.

Assigning category to test review: The assignment of category (i.e. fault or false-positive) to a test review is explained with the following example sentence which is tested against the feature set generated in Table 2. The test review is as follows:

"System variable is not tested".

In this example, total size of the feature set is 9 (6 from fault category and 3 from non-fault); various frequent occurring words in English such as 'the', 'in' and 'is' are removed as part of pre-processing task to create more informative and descriptive feature set (details are in section IV). Our proposed approach extracts all unique features from

TABLE 2. EXAMPLE OF FEATURE SET GENERATION

Reviews	Class	Feature set size
"working:1", "system:2", "heavy:1", "load:2", "not:2", "tested:2"	Fault	6
"Who:1", "opened:1", "gate:1"	False-positive	3

a training sentence and stores them along with their frequency of occurrence. The test sentence is then tested against the feature set developed during training to assign a final category. As seen, the features from an example test sentence, "system" and "tested" have higher frequencies in "fault" of Table 2, while none feature occurs in false-positive class. So, this test sentence is assigned fault class because more *informative features* in fault class are matched.

Feature Set Generation using Part Of Speech (POS) tags: Part of speech (POS) tags are grammatical tagging or word-category disambiguation to mark up a word in text corpus corresponding to its lexical categories. In this study, we are proposing our approach in conjunction with POS tags to perform preliminary analysis of classification results for inspection reviews and movie reviews. Gimpel et.al [19] listed in their research that in any unstructured text,

the most important POS tags are Nouns, Adjectives, Verbs, Adverbs and Determiners. Their claim is that "extracting only the important POS out of an unstructured sentence delivers apposite sense". So, we trained our classification model over feature sets obtained for each of these POS tags. The analysis is performed over 20 intervals (5% to 100%) of feature-set for each POS tag. During our analysis, we observed that both our model performed poorer when trained over determiners. The reasons for this poor performance was due to use of '*engish stopword*' that removed most of the determiners (like 'the', 'an', 'that' etc.) during preprocessing steps. Additionally, there were very low number of features generated during training on determiners for both the models (less than 50 in inspection trained and less than 100 in movies trained); so we removed determiners from our analysis. The experiment design, experiment procedure, research questions investigated are discussed in next section.

IV. EXPERIMENT DESIGN

In this study, we applied mining approach using supervised learning classifiers (trained on two different review sets) that was tested on reviews generated from inspection on NL requirements document and validated its ability to differentiate between true-faults and false-positives. In this experiment, we also analyzed the impact that size of '*feature set*' (threshold for features) had on accuracy and G-mean. This section provides the discussion about study design, study goal and methodology.

A. Experiment Methodology

Research Questions (RQs): Two major research questions investigated in this study are listed below:

RQ 1: *Does training our mining approach on movie reviews and using part of speech (POS) tags overcome class imbalance problem associated with inspection reviews?*

RQ 2: *What impact does the size of features set included in training model had on evaluation metrics (F-measure and G-mean) of validating reviews?*

Independent Variables*:* Following independent variables were manipulated during the study run:

a) *Type of classifiers:* Nine (9) different classifiers are used in this study (along with the learning family to which they belonged) are shown in Table 3. These classifiers were chosen for this study based on literature review on mining textual information [3-4, 7].

TABLE 3. TYPE OF CLASSIFIERS USED

1. Bayesian	2. Support Vector	4. Regression
Naïve Bayes	Linear SVC	Logistic Reg.
Multinomial NB	3. Ensemble	SGD
Bernoulli NB	Random Forest trees	5. Trees
	Extra Tree	Decision Trees

b) *Training models:* Two different domains (inspection and movie) were used to train our mining approach. Inspection model was trained on 920 while movie model was trained on 10662 reviews.

c) *Feature set size:* Total number of features were randomly divided into 20 equal intervals to understand the effect of features on the classification results.

d) *Part of Speech (POS) tag:* we performed analysis over X different POS tags. These POS tags are Nouns (N), Adjective (J), Verbs (V), Adverb (R) and Determiner (DT).

Dependent Variables: The following variables were identified:

– *G-mean (Geometric mean)* is geometric mean of precision and recall and is measured as square root of multiplication of precision and recall. For example, in the following confusion matrix in Table 4, the accuracy is 65% [(9+4)/20)*100] and G-mean is [i.e. geometric mean of precision (0.69) and recall (0.75)] is 0.72 [√(0.69x0.75)].

TABLE 4. EXAMPLE OF CONFUSION MATRIX

	Predicted	
Actual	Positive (Fault)	Negative (Non-fault)
Positive (Fault)	9 (TP)	3 (FN)
Negative (Non-fault)	4 (FP)	4 (TN)

– *Threshold Value* (M_{thv}) is the percentage of features needed to train the model to achieve higher G-mean.

Participating subjects: A total of 41 students from software engineering (SE) course at NDSU (27 under-graduates and 14 graduates) participated and performed an inspection on software artifact that generated reviews used in this study.

Artifacts: Artifacts involved in this study to generate reviews are:

a) Inspection reviews: The inspection reviews were obtained from an inspection of Parking Garage Control System (PGCS) document that was developed externally at Microsoft and seeded with 35 realistic faults. There were total of 857 inspection reviews during the inspection process with 656 labelled as non-faults and 201 reviews as fault category. Clearly, it is a class imbalance problem in which the minority class ('fault') is approximately one-third the size of majority class (labelled as 'non-fault').

b) Movie reviews: Movie reviews were obtained from open source [26] and were pre-classified into negative (used to train model for fault category) and positive (used to train model for non-fault) category. There were total of 10662 movie reviews and equally divided into positive and negative category.

B. Experiment Procedure

The experiment procedure consisted of 5 steps and its procedural steps are shown in Figure 1. The experiment was run on system with 64-bit core-i7 processor and 32GB main memory running python-3.4 64bit version. The experiment was build up over NLTK (Natural Language Tool Kit) python language processing package along with Scikit-learn, Numpy, Scipy and Matplotlib scientific packages to carry out complex calculations and to use in-build classification support. The installation and implementation details can be found online. The details of experiment steps are:

Step 1 and 2: Due to space restriction, details of step 1 and step 2 can be found in study conducted by goswami et. al. in [2].

Step 3—Pre-processing and development of training and test sets*:*

Pre-processing*:* The data was cleansed by removing frequent occuring words in English language (known as stopwords in NLTK). The stopwords are generally removed because these are the words that are common to both categories in binary classification and do not contribute in prediction. All words are reduced to their base form with the help of lemmatization and this was performed to avoid multiple repitition of single word in different vocabulary. E.g. 'better', 'best' and 'good' all have one common base 'good' and lemmatization reduce the word to its base. The preprocessing task for

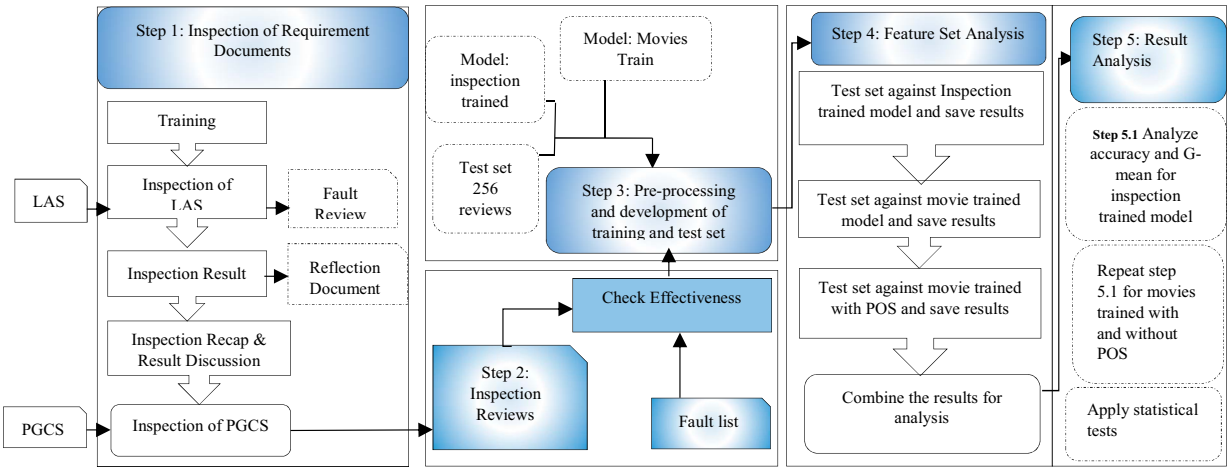


Figure 1. Details of experiment steps

POS tag is enhanced by filtering those words that doesn't belong to given POS under study. e.g if feature set is prepared for adjectives (J), then all other words that are not adjectives are removed.

Development of Training and Testing Sets: The total reviews generated from these 41 students were 857 (656 non-fault and 201 faults) in total and we used 70-30 split ratio to randomly select reviews for training and testing as shown in Figure 2. We performed five different analysis (Random Over-Sampling, Random Under-Sampling, SMOTE, AdaBoost and without sampling) of our training data to finalize the selection of the most accurate model. The model selection was performed using 10-folds cross-validation repeated 10 times. We selected 13 different classifiers in the beginning from 5 different families of classifiers (shown in Table 3) to perform cross-validation for each one of them. We then selected nine (9) that showed the most accurate cross-validation score. The mean of each repetition of 10-folds cross-validation is shown in Table 5 below. The score of cross-validation close to 1 signifies most accurate model and it can be seen from Table 5 that the score of SMOTE and ROS for each classifier was highest and almost identical while it was least for 'without sampling'. The class imbalance problem in training set was tackled by oversampling [6-8]. We chose ROS over SMOTE because of their performance for Random forest and Extra tree. Also, ROS was easy and less time consuming to implement. The other classifiers that we excluded from our analysis based on their poor performance during model selection (i.e. cross-validation) were Linear Regression, NuSVC, SVC and Gaussian Naïve Bayes.

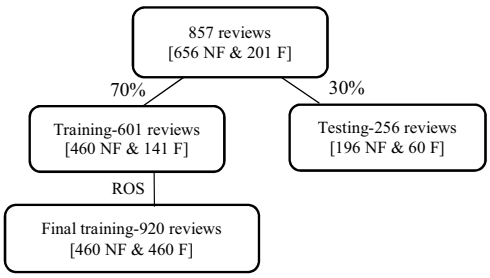


Figure 2 Training and testing set split

The oversampling of testing set is not performed to avoid redundant test reviews and the chance to avoid excessive misclassification error i.e. oversampled review in test set if classified into wrong category would results in higher misclassification error as the same prediction would be repeated for all other sampled copies of that review.

***Step 4 – Feature sets*:** Features in both models were varied from 5% to 100% and were divided into 20 equal intervals. Every review from both test sets was analyzed against both training models over variable size of feature sets. The interval of feature sets was chosen 5% because the 'inspection trained' model had only 368 features generated. Smaller feature interval would have added few new features to training model. similarly, the feature set was generated for five different part of speech tags for both training sets.

V. DATA COLLECTION

This section discusses the collection of data to answer the requirement questions developed in section III. Various metrics that were used in this study are described in Table 6. The accuracy shows the evaluation of classification model in predicting true-faults and non-faults while G-mean in this paper, shows the evaluation of classifier over minority class only (Table 6).

TABLE 5. CROSS VALIDATION RESULTS FOR MODEL SELECTION

Classifiers	Without Sampling	Ada Boost	RUS	ROS	SMOTE
Naïve Bayes	0.7211	0.717	0.6215	0.7373	0.7456
Multinomial Naïve Bayes	0.7984	0.723	0.6757	0.8247	0.8307
Bernoulli Naïve Bayes	0.7762	0.7502	0.6597	0.8304	0.8803
Decision Tree	0.7612	0.77	0.6373	0.8832	0.8402
Linear SVC	0.8130	0.816	0.7035	0.8948	0.9045
Random Forest	0.8044	0.8083	0.6702	0.9252	0.919
Extra Tree	0.8181	0.8191	0.6820	0.9511	0.9376
Logistic Regression	0.7943	0.751	0.6904	0.8520	0.857
SGD	0.7753	0.7323	0.6843	0.8713	0.8903

TABLE 6. DATA COLLECTION METRICS

Metric	Description
T_r	Total # of reviews fed to the model
C_{tf}	TF count in T_r
C_{fp}	FP count in T_r
TP	# of reviews classified as TFs by model
TN	# of reviews classified as false-positives by model
FP	# of non-fault reviews classified as faults by model
FN	# of fault reviews classified as non-fault by model
M_{thv}	Threshold %age of feature sets
Precision	TP/(TP+FP)
Recall	TP/(TP+FN)
F score	F-measure is harmonic mean of precision and recall; 2TP/(2TP+FP+FN)
G-mean	√(precision)x(recall)

This can be seen that the value of G-mean is predominantly contingent to actual faults found (TP), actual faults labelled as non-faults (FN) and actual non-faults labelled as faults (FP) by the classifier. G-mean and F-measure doesn't depend on correctly predicted majority class instances (TN) and that's why we have used this metric in our analysis that could present analysis for only true-faults predicted by the classification model.

VI. RESULTS AND ANALYSIS

This section report results regarding the application of two differently trained models ('inspection' vs. 'movie') at validating requirement reviews. The results compare G-mean and F-measure of test-set tested across varying features of each training model type. Figure 3 and 4 shows the outcome obtained by implementation of experiment procedure defined earlier in section III. The percentage of features varied (5% to 100%) are plotted along X-axis while G-mean and F-measure are plotted along Y-axis. The response variable (evaluation metrics) are observed for various POS explanatory variables over different feature set %age; W-POS, POS-J, POS-N, POS-R and POS-V stands for 'Without POS', 'POS adjective (J)', 'POS noun (N)', 'POS adverb (R)' and 'POS verb (V)' respectively. The results are organized across two RQs listed in section III.

RQ1: Evaluation of test set on trained models using POS vs. class imbalance problem.

We evaluated test set (containing 256 reviews with 196 FPs and 60 TFs) of requirement reviews using evaluation metrics (G-mean and F-measure) against both '*movies trained*' and '*inspection trained*' model. The purpose of this research question is to investigate if reviews from a semantically similar domain in NL context (e.g. movies in this case) could overcome class imbalance problem encountered while classification of inspection reviews. To test for any improvement, we analyzed and compared evaluation metrics of classification model when trained over reviews that are balanced w.r.t. outcome class distribution (belongs to different domain of movies) versus when trained over reviews that are imbalanced in class distribution (belongs to requirement reviews and are made balanced through ROS). The observations are taken for two scenarios (with and without the use of POS) and are shown in Figure 3 and 4.

Trained models without using POS (W-POS): The observations obtained in this scenario are taken by training both models over different feature set intervals (5% to 100%) but without considering specific part of speech tags i.e. without extracting features based on specific POS. The feature set size for W-POS is 368. The major observations from Figure 3 and 4 that are found are listed below:

- *Movies trained model*: G-mean and F-measure for W-POS explanatory variable does not show any stable trend.
- *Inspection trained model*: G-mean and F-measure shows negligible stable trend beyond 70% feature set interval.
- *Performance*: Movies trained under-performed for G-mean and F-measure over W-POS as compared to inspection trained.

Training models with POS: The observations obtained in this scenario are taken by training both models over different feature set intervals but considering four main part of speech tags namely Nouns (N), Adjective (J) and Verb (V). The size of feature set for Nouns is 281, Adjective is 120, Verb is 245 and Adverb is 56. The size of feature set for adverbs in inspection trained model being small was removed from our analysis. The major observations are listed below:

- *Movies Trained*: The most stable trend for G-mean and F-measure is shown only by POS-J and POS-N beyond a certain feature set percentage. The best performance is also shown by these two POS tags (N and J).
- *Inspection Trained*: Unlike movies trained, the inspection trained model behaved similar for all four POS tags for G-mean and F-measure beyond a certain feature set percentage. The trend of evaluation metrics for small %age of feature set is not as stable as shown by movies trained.
- *Performance*: The movies trained model performed better than inspection trained over POS-N and POS-J. It is also observed that the evaluation metrics show more stable trend in movies trained for POS-N and POS-J compared to any POS in inspection trained.

Implications: It is observed that the performance can be improved with the help of part of speech tags. It is seen that the POS tags that showed most improvement are Nouns (N) and Adjectives (J). It is also observed that the performance of both the models is improved using POS tags as compared to without using POS. It can be concluded that using POS approach and reviews from semantically similar domain can be substituted for class imbalance examples.

RQ2: Training Models versus Feature Set Threshold

In this requirement question, we investigated the threshold for features-to-include in training to achieve stable results. To do so, we analyzed each model over complete range of feature set (i.e. from 5%

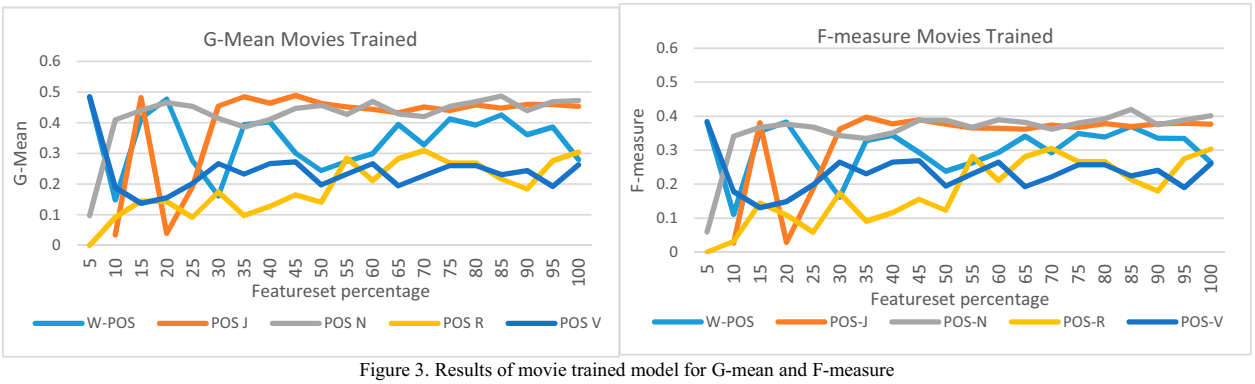


Figure 3. Results of movie trained model for G-mean and F-measure

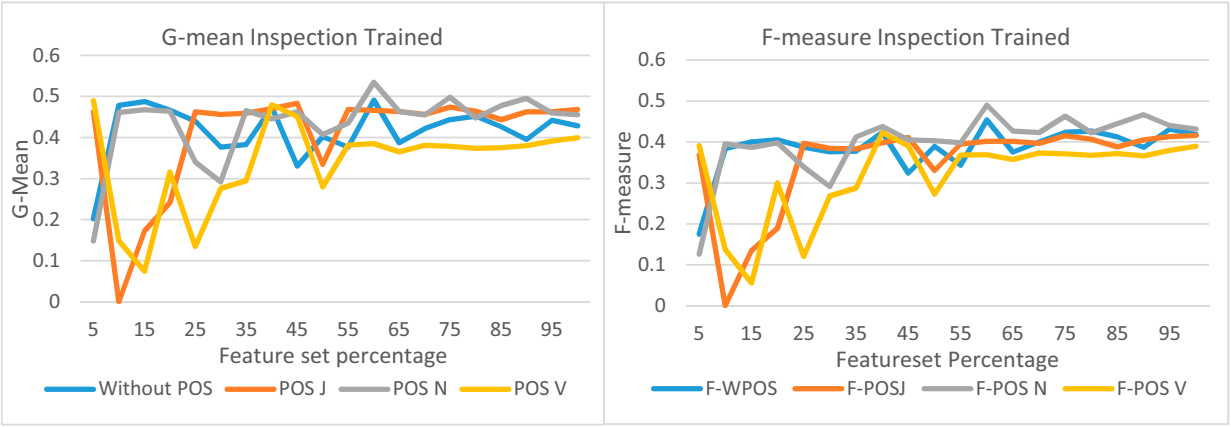


Figure 4. Results of inspection trained model for G-mean and F-measure

to 100% features). We divided the complete feature set into intervals of 5% forming 20 equally spaced intervals. Next, we observed G-mean and F-measure for most balance percentage interval beyond which the gain in performance becomes stable. The other reason towards observing for such a threshold interval is to minimize the tendency to misclassification error. The analysis is performed for each model against testing set using POS tags and various feature set intervals (Figure 3 and 4). The X-axis shows the percentage of features taken to train the model and Y-axis shows evaluation metrics. The analysis is performed separately for both the models i.e. for 'movies trained' (Figure 3) and for 'inspection trained' (Figure 4):

Trained models without using POS (W-POS): The major observations from results in the form of feature set threshold and evaluation metrics is discussed as follows:

- *Movies trained*: G-mean and F-measure for W-POS does not show a fixed feature set interval beyond which the metrics become stable.
- *Inspection trained*: G-mean and F-measure is unstable for most of the feature set intervals for W-POS. The graph shows negligible stableness beyond 65% feature-set interval.
- *Feature set threshold*: For movies trained, none feature-set %age interval could be found beyond which a stable G-mean and F-measure could be expected. For inspection trained model, it is observed that a stable trend for G-mean and F-measure can be found beyond 65% feature set interval for W-POS.

Trained models using POS (POS-J, POS-N and POS-V): The major observations from results in the form of feature set percentage and evaluation metrics is discussed as follows:

- *Movie trained*: As per observation, it is observed that a stable G-mean and F-measure is obtained for POS-N and POS-J beyond feature set percentage of 45%. For other two POS tags (V and R), the model does not show any stable threshold %age.
- *Inspection trained*: Figure 4 shows that the trend of G-mean and F-measure gets stable beyond feature set interval 65%.
- *Feature set threshold*: It is seen that for POS-N and POS-J, movies trained model shows most stable trend of evaluation metrics beyond 45% and 65% for inspection trained the stable trend is shown for all POS tags but the most accurate results were obtained for POS-N and POS-J.

Implications: It is observed that the performance of movies trained model is enhanced with the use of POS tags noun and adjectives and a similar enhancement with POS is seen for inspection trained model in comparison to without POS. The best feature set threshold for movies trained is 45% while for inspection trained is 65%. The difference in threshold values is significant because 45% of movies feature set is many times more than 65% of inspection train. The possible explanation for this threshold difference lies in the fact that movies reviews being from different domain compensate the gap with large number of features.

VII. Discussion of Results

The results presented in section V are discussed in this section. Because of space restriction, only graphical visualization of analysis is shown for movies trained model (Figure 5) and for only two part of speech tags (Nouns and Adjectives). The discussion is organized around two RQs; to aid discussion, Figure 5 (for movies trained model) plots G-mean and F-measure for feature set intervals (5% to 100%) over POS Nouns (N) and Adjectives (J). The X-axis in Figure 5 shows the percentage of feature sets included and Y-axis shows the evaluation metrics. We used R-square as a measure to evaluate model's capability. R-square value measures how well the observed values of response variables (evaluation metrics in this case) are predicted by the model. We used polynomial regression to fit the model's prediction to maximum polynomial degree of three. The R-square values for POS-J are fitted with polynomial regression of degree two while POS-N are fitted with polynomial degree of three. We didn't go fitting our model beyond polynomial degree of three because in such case, the model fits the data well but makes it very complex; also, it does not guarantee to approximate (fit) any new real-world relationship that the model has not seen earlier. Figure 5 is used to discuss results and their implications with the help of polynomial regression analysis.

***1) Discussion of RQ1 (Training Models Vs. POS Vs. Class Imbalance*)**: When the test set was analyzed against both models; similar results for evaluation metrics are obtained. The discussion involves the applicability of movie reviews to validate inspection reviews. The discussion also involves the significance of

TABLE 7. R-SQUARE AND P-VALUES FOR BOTH TRAINED MODELS

Trained models	Inspection Trained		Movies Trained	
	G-mean	F-measure	G-mean	F-measure
POS-J	0.3827	0.4751	0.6135 (2)	0.6431
P-Value	0.095836	**0.034265**	**0.004016**	**0.002223**
POS-N	0.3489 (3)	0.5455 (3)	0.5266 (3)	0.5797 (3)
P-Value	0.13163	**0.012856**	**0.01706**	**0.007384**
W-POS	0.1103 (3)	0.2997 (3)	0.1158 (3)	0.1001 (3)
P-Value	0.6434	0.199229	0.626847	0.674563

improvement over inspection trained model through POS approach. Table 7 lists R-square values along with corresponding P-values of both inspection trained and movies trained models. The significant P-values are made bold and the discussion is presented below:

a) Only those POS tags are compared in Table 7 that showed better performance than other-POS tags in section V. This left with POS-N and POS-J which are compared along with without POS (W-POS) for both the models.
b) W-POS (the traditional way without using POS) fails to show any significant R-square values of G-mean and F-measure for both the models. This could be because of the reason of imbalance between classification classes.
c) The R-squared values of POS-J and POS-N for G-mean and F-measure are improved for movies trained model and are proved significant with 95% confidence level with P-Value.

Implications: As seen in points discussed above, the values of R-square are improved for movies trained models. A significant improvement is also shown for G-mean value in movies trained. The improvement in G-mean values of movie reviews explain itself that the class imbalance problem is overcome upto an extent with the use of reviews from a semantically similar domain. The improvement over R-square values means that the model accurately predicts more percentage of data. This implies that the class imbalance problem can be resolved using reviews from a semantically similar domain.

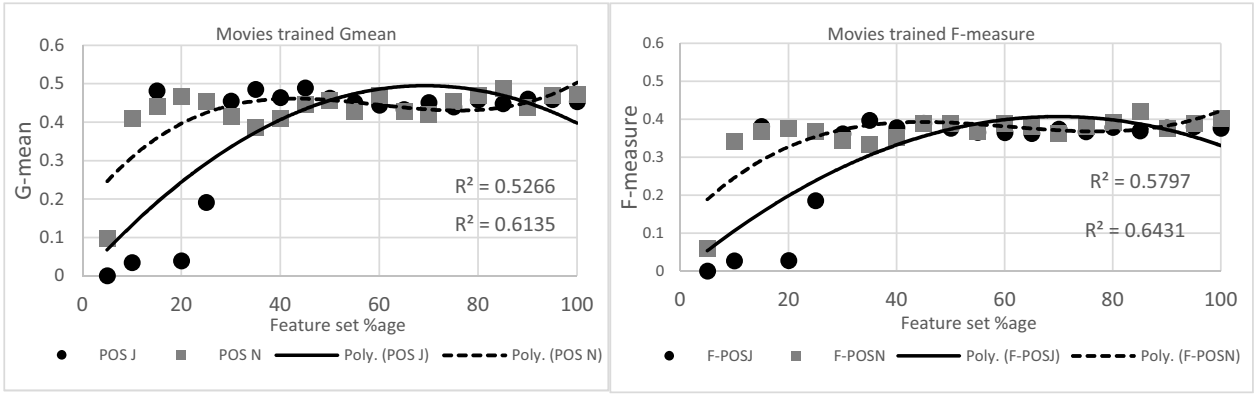
2) ***Discussion of RQ2 (Training Models Vs. Feature Set Threshold):*** In this section, the discussion is presented for the percentage of feature sets required to include while training in order to maximize accurate prediction. Because of space restriction the discussion is presented (in Figure 5) based on POS-N and POS-J for movie trained model only:

a) The R-square value for POS-J (0.6431) in Figure 5b shows that our model was able to predict 64.31% of the variance in response variable. The model with R-square value close to 1 is ideal but is difficult to construct when human responses (like reviews) are involved because of variance in reporting a similar thought in different way by different people.
b) The R-square values shown in Figure 5 are significant with P-value test at 95% confidence level.
c) In Figure 5 it is seen that for both POS-N and POS-J, the polynomial regression curve exactly fits the data points. POS-N data is exactly fitted by polynomial regression over interval (35% to 60%) while POS-J data is fitted over interval (40% to 90%) for both G-mean and F-measure.
d) From our discussion in section V, we observed that the stable behaviour in G-mean and F-measure is shown beyond feature set interval 45%. This can be seen from Figure 5 that beyond the interval value 45% , the polynomial curve exactly fits the data points for POS-N and POS-J.

Implications: From the above discussion, it can be concluded that the feature sets required to train a classifier over movie reviews require atleast 45% of feature sets. To maximize the prediction accuracy, the POS tags Nouns and Adjectives should be used. Similarly, when inspection trained model was analyzed for POS nouns and adjectives the threshold value for feature sets came out to be atleast 65%, beyond that the polynomial regression curve fitted the data more accurately.

VIII. THREATS TO VALIDITY

In this study, although full attention was given to remove as many validity threats as possible but there could be some remained. There could be an external validity threat in which our model uses 9 machine learning classifiers to differentiate between fault and false-positive category. Our belief is that if some other classifiers like neural network could be used then results may improve. Another external validity threat is that our test sets comprised of fault descriptions from graduate and undergraduate students; more professionally written descriptions if used to train models can also



a) *b)*

Figure 5. R-square and P-value analysis of movies trained data over POS tag Noun (N) and Adjective (J)

improve accuracy of models. Another threat which we believe could have existed is that we used movie reviews as substitute to inspection reviews. We need to check our models over some other publicly available rich text corpus. Some other approaches to perform sampling of reviews could also be analyzed to check for further improvements.

IX. Conclusion and Future Scope

Based on our results and discussion, we conclude that training over movie reviews produced better results and the model trained on movie reviews present most precise G-mean and F-measure score when they are trained on at least 45% of the total features generated; while for inspection trained model, this threshold value is observed to be beyond 65% of total features generated. It is also seen that misclassification rate is less at determined threshold value because of improved G-mean and F-measure. Using POS tags nouns and adjective along with reviews from semantically similar domain can address class imbalance problem. Our method also outperformed some sampling methods in predicting data with class imbalance problem. The difference in variable threshold percentage obtained by these two models is because of difference in total number of feature set generated for both models. Our automated approach could provide most accurate results to requirements author to help him save time and effort but it still needs to address misclassification errors; which is our future work to preserve true-faults from being misclassified and to provide more accurate post inspection decision support. Also, movie trained model has larger feature set (11318 in total) and 45% of 11318 is around 5000 features which is a lot more than inspection trained model. So, this opens another research investigation that could find a more semantically similar domain to train the model.

References

[1] J. Moeyersoms, E.J.D. Fortuny, K. Dejaeger, B.Baesens, D. Martens, “Comprehensible software fault and effort prediction: A data mining approach,” The journal of Systems and Software, Elsevier, Vol. 100, 2015.

[2] A. Goswami and G. S. Walia, "Teaching Software Requirements Inspections to Software Engineering Students through Practical Training and Reflection," in *Proceedings of the 123rd American Society of Engineering Education Conference on Computer Education ASEE 2016*, New Orleans, Louisiana,USA, 2016.

[3] I. L. Margarido, J. P. Faria, R. M. Vidal and M. Vieira, "Classification of defect types in requirement specification: Literature review, proposal and assessment," in *6th Iberian Conference on Informamtion Systems and Technologies (CISTI 2011)*, Chaves, Portugal, 2011.

[4] Yang, Cheng-Zen, et al. "An empirical study on improving severity prediction of defect reports using feature selection." Software Engineering Conference (APSEC), 2012 19th Asia-Pacific. Vol. 1. IEEE.

[5] L. Zhuang, F Jing, X. Zhu, “Movie Review mining and summarization,” in *Proceedings of the 15th ACM international conference on Information and knowledge management,* Arlington, Virginia, USA, 2006.

[6] Phung, Son Lam, Abdesselam Bouzerdoum, and Giang Hoang Nguyen. "Learning pattern classification tasks with imbalanced data sets." (2009).

[7] Nguyen, Hien M., Eric W. Cooper, and Katsuari Kamei. "A comparative study on sampling techniques for handling class imbalance in streaming data." Soft Computing and Intelligent Systems (SCIS) and 13th International Symposium on Advanced Intelligent Systems (ISIS), 2012 Joint 6th International Conference on. IEEE, 2012.

[8] Mustafa, Ghulam, et al. "Solving the class imbalance problems using RUSMultiBoost ensemble." Information Systems and Technologies (CISTI), 2015 10th Iberian Conference on. IEEE, 2015.

[9] Qin, Sijun, et al. "Feature selection for text classification based on part of speech filter and synonym merge." Fuzzy Systems and Knowledge Discovery (FSKD), 2015 12th International Conference on. IEEE, 2015.

[10] Wang, Meng, Lanfen Lin, and Feng Wang. "Improving short text classification through better feature space selection." Computational Intelligence and Security (CIS), 2013 9th International Conference on,.

[11] S. Qin, J. Song, P. Zhang, Y. Tan, “Feature Selection for Text Classification Based on Part of Speech Filter and Synonym Merge,” 12th International Conference on Fuzzy Systems and Knowledge Discovery, 2015.

[12] C. Yang, C. Hou, W. Kao and I. Chen, “An Empirical Study on Improving Severity Prediction of Defect Reports using Feature Selection” 19th Asia-Pacific Software Engineering Conference, 2012, pp. 240-249.

[13] S. Bahassine, A. Madani, M. Kissi, “An improved Chi-square feature selection for Arabic text classification using decision tree,” 11th international conference on Intelligent Systems: Theories and Applications (SITA), 2016.

[14] Kubat, Miroslav, and Stan Matwin. "Addressing the curse of imbalanced training sets: one-sided selection." ICML. Vol. 97. 1997.

[15] Bavota, Gabriele. "Mining Unstructured Data in Software Repositories: Current and Future Trends." Software Analysis, Evolution, and Reengineering (SANER), 2016 IEEE 23rd International Conference on. Vol. 5. IEEE, 2016.

[16] A. Bosu, M. Greiler and C. Bird, "Characteristics of Useful Code Reviews: An Empirical Study at Microsoft," in *12th Working Conference on Mining Software Repositories*, Florence, Italy, 2015.

[17] N. Chen, J. Lin, S. Hoi and X. Z. Xiao, "AR-Miner: Mining Informative Reviews for Developers from Mobile App Marketplace," in *ICSE 2014: 36th International Conference on Software Engineering*, India, 2014.

[18] A. X. Agarwal, I. Vovsha, O. Rambow and R. Passonneau, "Sentiment Analysis of Twitter Data," in *Processings of the Workshop on Languages in Social Media*, 2011, pp30-38.

[19] K. Gimpel, N. Schneider, B. O'Connor et.al, "Part of Speech Tagging for Twitter: Annotation, Features, and Experiments," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics:shortpapers*, PortLand, Oregon, june, 2011.

[20] O. Abdelwahab, M. Bhagat, C.J. Lowrance and A. Elmaghraby, “Effect of training set size on SVM and Naïve Bayes for Twitter sentiment analysis, IEEE international symposium on signal processing and information technology, 2015.

[21] K. K. Marri, "models for evaluating review effectiveness," in *3rd Annual International Software Testing Conference*, India, 2001.

[22] H. Arora, V. Kumar, R. Sahni, “Study of bug prediction modeling using various entropy measures- a theoretical approach,” 3rd international Conference on Reliability, Infocom Technologies and Optimization, 2014.

[23] Carver, J.: ‘The impact of background and experience on software inspections’, Empirical Software Engineering, 2004, 9, (3), pp. 259-262.

[24] A. Bacchelli and C. Bird, "Expectations, Outcomes and Challenges of Modern Code Review," in *International Conference of Software Engineering* , San Francisco, USA, 2013

[25] L. V. G. Carreno and K. Winbladh, "Analysis of User Comments: An Approach for Software Requirements Evolution," in *ICSE 2013: International Conference of Software Engineering*, San Francisco, USA.

[26] “short reviews”, python programming, [online]. Available: pythonprogramming dot net /static /downloads/short reviews [Accessed 18 06 2017].

[27] Anu, V., Walia, G., Hu, W., Carver, J., and Bradshaw, G. “Using a Cognitive Psychology Perspective on Errors to Improve Requirements Quality: An Empirical Investigation”, 27th IEEE International Symposium on Software Reliability Engineering. ISSRE 2016. Ottawa, Canada.