## Background

## Literature Review

Classification of software bugs using bug attribute similarity (CLUBAS) is proposed by N. K. Nagwani et. al. [1]. The proposed work can be divided into five steps- data pre-processing, text clustering, frequent term calculations, taxonomic terms mapping techniques and performance evaluation. In the first step data are pre-processed by eliminating stop words and applying stemming over the textual bug attributes. In the second step text clusters are created using textual similarity between the attributes summary and description for each pair of bugs. In the third step cluster labels are generated calculating the frequent and meaningful terms from each cluster text data and assign them to that cluster. In the fourth step cluster labels are mapped against the bug taxonomic terms to identify appropriate categories of the clusters. In the fifth step performance is evaluated by calculating accuracy, precision, recall and F-measure.

The algorithm CLUBAS shows stability and performs better than Classification using Clustering (CC), Support Vector Machine (SVM) and J48 when this algorithm is implemented on the Android bug repository. It maintains the F-measure value more than 0.9 for each experiment using different number of samples. When the precision, recall and F-measures are important CLUBAS gives the better and stable results irrespective of number of samples and software bug repositories. But both Naïve Bayes (NB) and Naïve Bayes Multinomial (NBM) performs better in terms of accuracy than CLUBAS.

To improve the performance of CLUBAS advance text pre-processing techniques can be implemented to optimize the clustering and classification work and also modern text clustering and classification can be implemented.

Recent approaches in automatically classifying bug reports as bug and non-bug are based on text mining. Antoniol et al. [2] have investigated the automatic classification of bug reports by utilizing conventional text mining techniques, which demonstrated the feasibility. They have used title and description of bug reports.  They have collected about 1800 bug reports from Mozilla, Eclipse and Jboss. The approach consists of three pipeline phases. First, they have manually classified the bug reports as bug or non-bug. Second, the classifier is trained by the labelled data. Third, they have predicted the bug reports of test data using the trained classifier. The precision and recall was about 72-75% and 75% respectively.

## References

1. Nagwani, Naresh Kumar, and Shrish Verma. "CLUBAS: an algorithm and Java based tool for software bug classification using bug attributes similarities." *Journal of Software Engineering and Applications* 5.06 (2012): 436.
2. Antoniol, Giuliano, et al. "Is it a bug or an enhancement?: a text-based approach to classify change requests." *Proceedings of the 2008 conference of the center for advanced studies on collaborative research: meeting of minds*. ACM, 2008.