ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 20, April 2016

An Approach to Calibrate Model Parameters in the Context of Source Code Document Clustering with Topic Modeling for Feature Extraction

Baby Rani A.S.[1], Nadira Banu Kamal A.R.[2]

[1]Assoc. Prof. in Computer Science, Sri Meenakshi Govt. Arts College for Women(Autonomous), Madurai, Tamil Nadu, India

[1]asrani18@yahoo.com

[2]Assoc. Professor & Head, Dept. of Computer Science , Thassim Beevi Abdul Kader College for Women(Autonomous), Kilakarai, Ramanathapuram, Tamil Nadu, India

[2]nadirakamal@gmail.com

***Abstract* -Feature Extraction in the context of Software Engineering (SE) tasks can be defined as identifying or locating the relevant source code modules related to the queries for the purpose of bug localization, traceability, finding software evolution etc. In this paper, we made a study on how effective be the Latent Dirichlet Allocation (LDA) modeling for clustering the source code modules. We have considered the SE task – Feature Extraction for our study. LDA a generative probabilistic method provides a model of the textcorpus that depends on its configuration parameters. We applied dynamic programming approach to calibrate the LDA parameters and its efficiency is evaluated with cluster analysis metric. Our work contributes a fast and simple method to fine tune LDA parameters (a near optimal solution) that provides results comparable with that of the work by other researchers.**

Keywords - **Latent Dirichlet Allocation, Clustering, Feature Extraction, Silhouette Coefficient**

I.INTRODUCTION

Recently, topic modeling [2] is used in various Software Engineering tasks, a method applied in Natural Language Processing[13] to elicit the semantics of the words with the contexts rather referring Thesaurus or Dictionary. It is applied with the principle that much information is hidden in the comments and identifiers that can be leveraged to group the similar documents which can simplify the source code search. But its effectiveness depends on the model parameters of LDA [3], that is having a serious impact on the clusters of related documents.

Girish Maskeri et al. in [6] has applied LDA for mining topics in the large business application. They found automatically generated optimal number of topics using log likelihood method gives false positives and also stated choosing model parameters is critical in applying the topic model. Software cost models can be learned using Bayesian Reasoning methods[7]. It is applied in Software Testing [8] in which researchers, besides discussing the challenges in applying this probability based technique in generalization, emphasized the applicability of this method for SE tasks. Software Evolution has been analyzed by making the study on the topics focused over a period[12]. Thomas et al. validated the use of topic modeling for software evolution by investigating the entropy of the topic metrics with the changes in source code [14]. Bugs reports document is

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 20, April 2016

analyzed with LDA in [9],[15] to locate the source code prone to error has been proved to be efficient when compared to other IR methods.

II.BACKGROUND

A.LDA

Latent Dirichlet Allocation modeling [2] is a generative probabilistic model of a corpus. The basic idea is that documents are represented as random mixtures over latent topics where each topic is characterized by a probability distribution over words.

LDA assumes the following generative process for each document **w** in a corpus D:

1. Choose $N \sim \text{Poisson}(\xi)$.
2. Choose $\theta \sim \text{Dir}(\alpha)$.
3. For each of the N word w_n ,
   (a) Choose a topic $z_n \sim \text{Multinomial}(\theta)$
   (b) Choose a word w_n from $p(w_n | z_n, \beta)$, a multinomial probability conditioned on the topic z_n.

- **w** represents a document (a vector of words) where $w = (w_1, w_2, \ldots w_N)$
- α is the parameter of the Dirichlet distribution, technically $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_k)$, but unless otherwise noted, all elements of α will be the same.
- z represents a vector of topics, where if the ith element of z is 1 then w draws from the ith topic.

β is a k x V topic by termprobability matrix for each topic (row) and each term(column), where $\beta_{ij} = p(w^i = 1 | z^k = 1)$

B. Clustering

We can define a cluster as a set of documents inwhich each document is closer (i.e., closer in a vector space) to every other document in the cluster, and it is fartherfrom any other document from the other clusters.Collections of natural languagedocuments are usually heterogeneous, meaning that documentscan contain information related to multiple topics. In sourcecode artifacts, heterogeneity is not always present, especiallywhen considering single classes. In [1] it is stated that a class is acrisp abstraction of a domain/solution object, and should havea few, clear responsibilities and hence, software documents shouldbe clustered considering only the dominant topic, assumingthat each document is related to only one specific topic. But experiment shows a sole dominant topic is having a little contribution towards class or method cohesion property. Different LDA configurations provide different clusteringmodels of the documents. However, not all clustering modelsthat can be obtained by configuring LDA are good. But surely the clustering method applied is having an impact on grouping related documents. We applied the basic k-Means clustering method. In K-Means clustering, given the no. of clusters K, a document will be assigned to a cluster if the mean distance of the cluster is minimal with this document inclusively.

C. Cluster Metric

Thereare two basic ways to evaluate the quality of a clusteringstructure: *internal criteria*, based on similarity/dissimilaritybetween different clusters and *external criteria*, which usesadditional and external information. Since the internal criterion does notrequire any manual effort and it is not software engineeringtask dependent, in our work we use the *internal criteria* formeasuring the quality of clusters. More specifically, internal quality metrics applied in [1]: *cohesion* (similarity),which determines how closely related the documents in a clusterare, and *separation* (dissimilarity), which determines howdistinct (or well-separated) a cluster is from other clusters.Since these two metrics are contrasting each other, we use apopular method for combining both cohesion and separationin only one scalar value, called *Silhouette coefficient*. TheSilhouette coefficient is computed for each document using theconcept of centroids of clusters. Formally, let C be a cluster;its centroid is equal to the mean vector of all documentsbelonging to C: Centroid(C) = $\sum_{d_i \in C} d_i$ /|C|.

Starting from the definition of centroids, the computation ofthe Silhouette coefficient consists of the following three steps:

1) For document d_i, calculate the maximum distance fromd_i to the other documents in its cluster. We call thisvalue $a(d_i)$.
2) For document d_i, calculate the minimum distance fromd_i to the centroids of the clusters not containing d_i. Wecall this value $b(d_i)$.
3) For document d_i, the Silhouette coefficient $s(d_i)$ is:

$$s(d_i) = \frac{b(d_i) - a(d_i)}{\max(a(d_i), b(d_i))}$$

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 20, April 2016

The value of the Silhouette coefficient ranges between -1 and 1. A negative value is undesirable because it relates to the case where $a(d_i) > b(d_i)$, i.e. a document in one cluster is closer to a document belonging to another cluster and hence it implies poor clustering.

For measuring the distance between documents we used the squared Euclidean distance. The overall measure of the quality of clustering $C=\{C_1,C_2,\ldots C_k\}$ can be obtained by computing themean Silhouette Coefficient of all documents as follows.

$$S(C) = \frac{1}{n}\sum_{i=1}^{n} s(d_i)$$

This measure is used to evaluate the quality of LDA configuration.

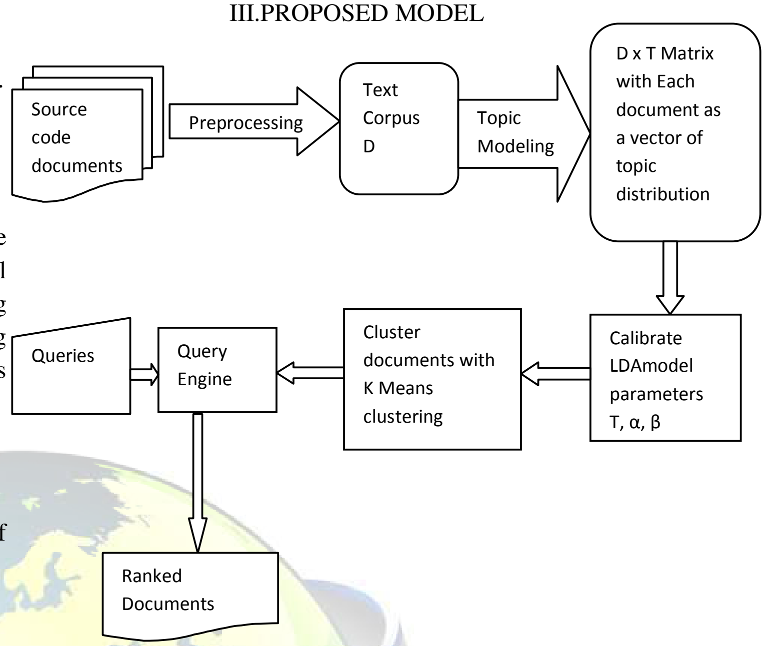
D. Feature Extraction

Features are framed as a query which can be just a set of keywords to search or the bug report or excerpts from the comments of source code document that depends on SE task undertaken. We used the set of keywords to search as a query Q. In LDA, similarity (Sim) between a document d_i and a query Q is computed as the conditional probability of the query given the document [14]:

$$Sim(Q,d_i) = P(Q|d_i) = \prod_{q_k \in Q} P(q_k \mid d_i)$$

where q_k is the k^{th} word in the query Q. Documents are ranked on the above probability measure and hence a more relevant document is the one with the high probability value. Also, the top ranked documents are examined for its cluster distribution. If all the top ranked documents are found within few clusters that indicates LDA model parameters are good.

III.PROPOSED MODEL



In the field of Text Mining, Natural Language Processing and Machine Learning, LSI and pLSI [4],[5],[8],[19]has been used in the research community. Recently, LDA method is applied to different applications [10],[11],[15],[16] for unsupervised machine learning and clustering. The experimental results in [1] caution the researchers that the configurational parameters of LDA is having a significant impact on the model derived and hence there is a need for finding the near optimal LDA parameters to effectively use for text mining. Research studies show that the best configuration parameters for one corpus is not applicable for the other. Poshyvanyk et. al in their work on the use of topic models [1], they compared different approaches like combinatorial, genetic algorithm(GA), source locality heuristics for finding near optimal configuration parameters. They are able to arrive at good configuration parameters with GA that is validated with Combinatorial results which is an exhaustive method. LDA_GA finds optimal parameters with fewer iterations when compared to combinatorial method. In this paper, we tried an alternative approach, a stepwise refinement of each of the parameters [α, β, T] where α , β are defined in section 2.1 and T is the number of Topics , by applying heuristics in multiple stages.

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 20, April 2016

A.LDA Parameter Calibration

To find the near optimal LDA configuration parameters, we applied dynamic programming approach that works in multiple stages. In this approach, a problem can be split up into a reasonable number of subproblems in such a way that a suboptimal solution can be retained so that if a better solution is not arrived in later stage, this solution can also be used.

In initial stage, fixing the value for number of topics T and hyperparameter α by applying heuristics , T = 400 or 500 that is mentioned in [13] as for any large text document , maximum of 400 or 500 topics will suffice to represent the document as a probability distribution of topics. The value of α which is the confidence level or the gain factor for sampling the input documents can be set to 50/T that is referred in [18].

In Stage I, fixing the parameters T and α, the value of hyperparameter β can be varied between 0.01 and 0.09 in steps of .01 . The higher the value of β, the coarser the word distribution in a topic. The mean efficiency of Silhouette Coefficient metric S(C) is computed for each set of configuration that ranges between -1 and 1.The best two parameter settings of β will be used in the next stage.

In Stage II, for each of the best two β values, the value of α is taken above and below the earlier fixed value (50/T). Again the mean efficiency is computed and best four results are considered for the next stage.

In Stage III, the property of topic dominance within the document and topic prevalence in the corpus is used as a decision criteria for reducing the number of topics.

Definition I : Let θ be the topic-by-document a T x n matrix generated by a particular LDA configuration P = [T, α , β] where n is the number of documents . A topic is said to be more dominant if and only if its probability value for all documents $\sum_{j=1}^{n} \theta_{ij}$ for i = 1..T, is above the mean value for the entire corpus.

Definition II : Let θ be the topic-by-document a T x n matrix generated by a particular LDA configuration P =[T, α , β] , d_i be the i^{th} document and f_i be the no. of topics above a threshold value in document d_i. A document $d_i$is said to have more prevalent topic if f_i is above the median value(m) for the entire corpus where m = median(f_i) for i = 1..n.

Based on these two definitions find topics that are a) more prevalent and more dominant b) more prevalent and less dominant c) less prevalent and more dominant and d) less prevalent and less dominant.

Our experimental study shows a more dominant and less prevalent topic is insignificant in clustering the documents. Hence compute the count (t) of such topics and reducing number of Topics T by t , LDA parameters can be reconfigured.Note, minimizing the number of Topics T reduces the dimensionality of the topic-by-document matrix M that leads to faster execution of any function that uses the values of M.

LDA model for source code documents is constructed as follows :

1. Split the class methods as separate documents.
2. Extract comments and identifiers from the source code documents.
3. Remove stop words and do stemming with Porter's algorithm[17].
4. Determine the term frequency of each word or term in the document that is stored as a sparse matrix in a file *f*.
5. Fast Gibbs Sampling [18] method is used to sample the documents with Latent Dirichlet distribution that results in wp (word-by-topic) matrix, dp (document-by-topic) matrix of probability distributions and a vector Z that assigns the topic to the vocabulary of the words in the document in the sequence in which it occurs.
6. Cluster the documents using K-Means clustering with cluster size 50 and

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 20, April 2016

evaluate mean Silhouette Coefficient S(C).

To calibrate the LDA parameters, our dynamic programming method is applied for step 5. Our experiments are able to arrive at the optimal mean efficiency of the clusters obtained by Thomas et. al in their work [21] and Poshyvanyk et. al in [1] .

B. Feature Extraction

Search Queries in the form of a string of keywords is input to the query engine to extract the relevant documents. We can measure the cosine distance between the vector of Document d and Query q. In applying LDA model, we can compute the conditional probability p(q|d) using the dp - document-by-topic matrix M obtained from the LDA model. We used the latter method and computed $p(q|d_i)$ for all documents i = 1..n and ranked the documents in the order of highest probability. Also top 20 ranked documents are examined for its distribution in various clusters. Fewer clusters implies that the LDA Model is more promising. Hence the distribution of more relevant documents in various clusters is measured as

E = Number of Clusters / Number of Documents

and the Mean Distribution Factor is the the average of E over all the queries. The Search Efficiency is inversely proportional to E.

IV.CASE STUDY

We have taken the open source software ArgoUML and jEdit for our study in order to validate our results with the related work done in [1].

The characteristics of the System under study are

TABLE : 1

System	KLOC	Classes	Methods	Queries
ArgoUML	149	1439	10999	91
jEdit	104	503	6347	150

Source code in the preprocessed form provided by Poshyvanyk et al.in the link[1] is used .

Fast Gibbs Sampling LDA topic model [18] is applied to our System for modeling the source code corpus. Our proposed algorithm is implemented in MatLab and the results are presented in the following section.

System	**LDA Parameters**			**Cluster Size**	**Mean Silhouette coefficient**	**Mean Cluster Distribution factor**
	T	**α**	**β**			
ArgoUML	300	.125	.05	50	.3694	0.4
jEdit	400	.5	.5	50	.3514	0.65

V.RESULTS & DISCUSSION

The methods explainedabove in section III are applied for the system under study and the optimal results are tabulated in Table II.Figure I(a) shows the histogram of no. of documents in every cluster for ArgoUML. The distribution of the most relevant documents (top 20 documents) in various clusters for each query is computed and its average variability is shown as boxplot in Figure I(b). It shows 75% of the query results lie below the Mean Cluster Distribution factor . For the system jEdit the results are shown in Figure II(a) and Figure II(b). For jEdit, the variability is slightly higher still the highest distribution factor is less than 0.8 and 50% of the query results arebelow the Mean Cluster Distribution Factor.

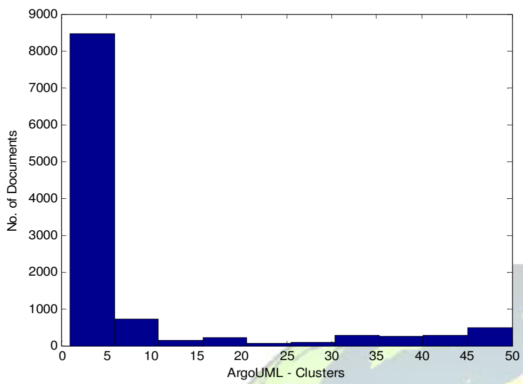
[1]http://www.cs.wm.edu/semeru/data/benchmarks/

These results are comparable to the results obtained in [1] & [21]. The number of iterations to calibrate the LDA parameters by applying the dynamic programming method is very few as compared to the combinatorial method and LDA_GA method applied in [1].

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 20, April 2016

TABLE : 2

FIGURE I(a)



FigureI(a)ArgoUML-Document Distribution in different Clusters

FIGURE I(b)

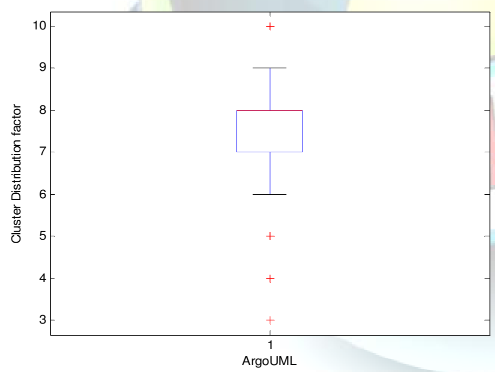


Figure I(b) ArgoUML - Cluster Distribution Factor of Matching Documents for the Queries

FIGURE II(a)

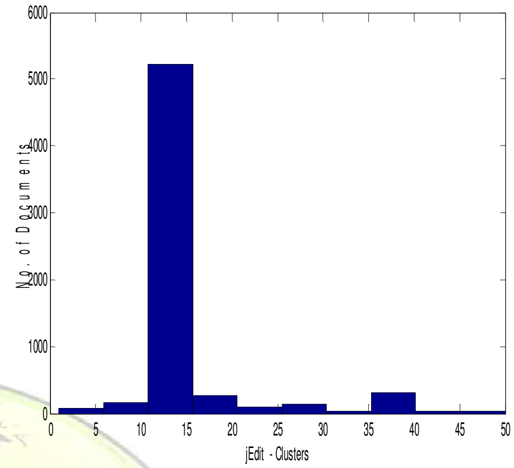


Figure II(a) jEdit - Document Distribution in different Clusters

FIGURE II(b)

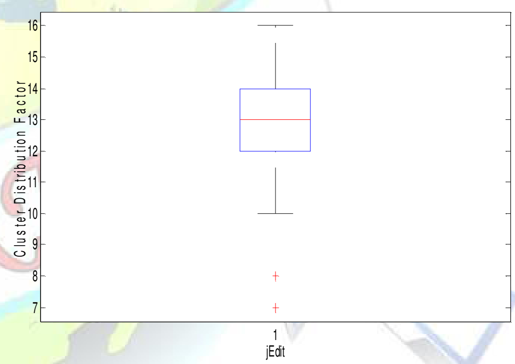


Figure II(b) jEdit - Cluster Distribution Factor of Matching Documents for the Queries

VI.CONCLUSION

Literature study reveals LDA method is getting popular in information retrieval in the unsupervised machine learning approach and it has been applied for various SE tasks. But its efficiency lies with the modeling parameters. As stated in [1], without proper calibration it leads to poor results and also it varies from system to system. Hence, an automation is required to find the model parameters and our method suggests an easy , simple and efficient approach in finding the

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 20, April 2016

near optimal solution. Its efficiency has also been validated with the Feature Location SE task.

In our future work, we will validate the applicability of LDA with other SE tasks. As far as source code is concerned, its structural information [20] also is equally important in feature location. We will try to apply other methods and compare its performance with IR method for feature extraction.

REFERENCES

[1] Annibale Panichella, Bogdan Dit, Rocco Oliveto, Massimilano Di Penta, Denys Poshynanyk, Andrea De Lucia, "How to Effectively Use Topic Models for Software Engineering Tasks? An Approach Based on Genetic Algorithms ",
IEEE ICSE 2013, pp. 522-531

[2] D.M. Blei, A.Y. Ng, and M.I. Jordan. "Latent Dirichlet Allocation", The Journal of Machine Learning Research, 3:993–1022, 2003.

[3] Colorado Reed " Latent Dirichlet Allocation: Towards a Deeper Understanding", January 2012

[4] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. "Indexing by Latent Semantic Analysis " , Journal of the American Society for Information Science, 41(6):391–407, 1990.

[5] Mark Girolami , Ata Kab´an, " On an Equivalence between PLSI and LDA "ACM SIGIR 2003.

[6] G. Maskeri, S. Sarkar, and K. Heafield. "Mining Business Topics in Source Code using Latent Dirichlet Allocation," In Proc. 1st India Software Engineering Conference, Hyderabad, India, February 2008, pp. 113-120

[7] Akbar Siami Namin , Mohan Sridharan "Position Paper: Bayesian Reasoning for Software Testing", Intl. Workshop on Future of Software Engineering , FoSER 2010.

[8] D. Poshyvanyk, Y. Gael-Gueheneuc, A. Marcus, G. Antoniol, and V. Rajlich, "Feature location using probabilistic ranking of methods based on execution scenarios and information retrieval," IEEE Trans. on Softw. Eng., vol. 33, no. 6, pp. 420–432, 2007

[9] S. K. Lukins, N. A. Kraft, and L. H. Etzkorn, "Bug localization using Latent Dirichlet Allocation," Information and Software Technology, vol. 52, no. 9, pp. 972–990, 2010.

[10] R. Oliveto, M. Gethers, D. Poshyvanyk, and A. De Lucia, "On the equivalence of information retrieval methods for automated traceability link recovery," in Proc. of the 18th IEEE International Conference on Program Comprehension (ICPC'10), pp. 68–71

[11] Taşcı S. , Güngör T. , " LDA-based Keyword Selection in Text Categorization ", 24th International Symposium on Computer and Information Sciences, 2009. ISCIS 2009

[12] A.J. Hindle, M. W. Godfrey, and R. C. Holt, "What's hot and what's not: Windowing developer topic analysis," in Proc. of the 25th IEEE International Conference on Software Maintenance (ICSM'09), Edmonton, Canada, September 20-26, 2009

[13] T. L. Griffiths and M. Steyvers, "Finding scientific topics", The National Academy of Sciences, Vol.101, 2004, pp.5228-5235.

[14] Thomas, S.W. , Adams, B. , Hassan, A.E. , Blostein, D. , "Validating the Use of Topic Models for Software Evolution "10th IEEE Working Conference on Source Code Analysis and Manipulation (SCAM), 2010, pp. 55-64

[15] S. K. Lukins, N. A. Kraft, and L. H. Etzkorn. "Bug localization using latent Dirichlet allocation ", Information and Software Technology, 52(9):972–990, 2010.

[16] M.Revelle, B.Dit and D.Poshyvanyk, " Usingdatafusionandweb mining to support feature location in software", In Proceedings of the 18th International Conference on Program Comprehension, pages 14–23, 2010.

[17] GibbsLDA++. http://gibbslda.sourceforge.net/.Int. Conf. on Research and Development in Information Retrieval, Toronto, Ontario, Canada, July 2003, pp. 433-434.

[18] The Porter Stemming Algorithm,
http://tartarus.org/~martin/PorterStemmer/.

[19] D. Poshyvanyk, Y.G. Guéhéneuc, A. Marcus, G. Antoniol, and V. Rajlich, "Combining Probabilistic Ranking and Latent Semantic Indexing for Feature Location", In Proc. 14th IEEE Int. Conf. on Program Comprehension, Athens, Greece, June 2006, pp. 137-148.

[20] D. Poshyvanyk and A. Marcus, "Combining Formal Concept Analysis with Information Retrieval for Concept Location in Source Code", In Proc. 15th IEEE Int. Conf. on Program Comprehension, Banff, Alberta, Canada, June 2007, pp. 37-48.

[21] Thomas S.W. , Nagappan M. ,Blostein D., and Ahmed E. Hassan "The Impact of Classifier Configuration and Classifier Combination on Bug Localization " , IEEE Trans. on Softw. Eng., vol. 39, no. 10, pp. 1427–1443, 2013.