## A NOVEL STUDY FOR SUMMARY/ATTRIBUTE **BASED BUG TRACKING CLASSIFICATION USING** LATENT SEMANTIC INDEXING AND SVD IN DATA MINING

Ketki<sup>1</sup>, Mr. Sanjay Kumar<sup>2</sup>, Mr. RajKumar Singh Rathore<sup>3</sup>

<sup>1</sup>PG Scholar, Galgotias College Of Engineering & Technology, Greater Noida(up), (India) <sup>2</sup>Assistant Professor, <sup>3</sup>M.Tech Coordinator, Dept of CS/IT, Galgotias College of Engineering & Technology, Greater Noida(up), (India)

## ABSTRACT

This paper presents LLatent Semantic Indexing (LSI) method for learningBug tracking concepts in document data. Each attribute in a vector provides the mark of participation of the document in data or term in the parallel concept. The objective to describe the concepts summary based, but to be capable to signify the documents and relations in a combined way for showing document-similarity, document-term, and term-term similarities or semantic relationship. Many technique for the implementing our research i.e. NLP, STEMMING, LUD, &SVD to the relevant similarity like bug report bug title, report summary etc., sinceevery bug report, and mined developer's name who fixed the bug reports from the developers activity data. We processed the mined textual data, and got the term-to-document matrix. Our proposed model different machine learning methods for on the basis of summary and attribute based classification of bug reports which produce the outlier bugs from large amount of data that improved accuracy and efficiency bug classification and relevant base bug indexing which is similar in meaning by using LSI and SVD Technique.

Keyword: BTS, LSI, SVD, LUD etc

## **I. INTRODUCTION**

A bug tracking system is used for justification and sharing of bug reports to the maximum appropriate designers. An LSI based bug trackingscheme may decrease the software preservation time and advance its value by correct and timely task of new bug reports to the suitable reporters.

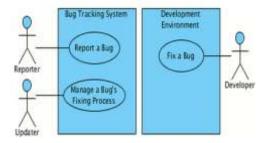


Figure 1: Diagram Shows Some use Cases Related to Bug Tracking and Fixing

# International Journal of Advanced Technology in Engineering and Sciencewww.ijates.comVolume No 03, Special Issue No. 01, May 2015ISSN (online): 2348 - 7550

In this paper, we present the current techniques overdue an LSI bug tracking system, which is based on the labeling of summary bug reports. In order to gets an attribute bugtracking system we used these techniques and performed comparative experiments.

Latent Semantic Analysis arose from the problem of how to find relevant documents from search words. The fundamental difficulty arises when we compare *words* to find relevant documents, because what we really want to do is compare the *meanings or concepts behind the words*. LSA attempts to solve this problem by mapping both words and documents into a "concept" space and doing the comparison in this space.

Most NLP applications such as information extraction, machine translation, sentiment analysis and question answering, require both syntactic and semantic analysis at various levels. The motivation for LU decomposition is based on the observation that systems of equations involving triangular coefficient matrices are easier to deal with.

## **1.1 Indexing for Documents**

Indexing is an essential part of the IR systems for two reasons. First, it optimizes the query performance and improves the respond times considerably by storing terms in an inverted file structure.

### **1.2 Searching**

In this phase, the query terms are searched against the inverted index. All the documents that contain the occurrences of the query terms are retrieved. Depending on the application, the retrieval can be done even for the partially matched documents.

## 1.3 Ranking

The documents retrieved in the previous step are given scores according to the matching quality between the query terms and the documents. The documents are sorted according to this score, so that the most relevant documents are presented to the user on top of the retrieval list. Ranking process is highly dependent to the IR model.Semantic analysis at various levels. Traditionally, NLP research has focused on developing algorithms that are either language-specific and/or perform well only on closed-domain text.

## **II. LITRATURE REVIEW**

Literature reviews are the maximum basic, yet very significant concept to set a theoretical basis. Their quality and usefulness greatly depends on the literature research process:-

Turney [2, 9] proposed the Latent Relational Analysis (LRA) by covering VSM-based methods in three ways: a) lexical patterns are mechanically extracted from a corpus, b) the Singular ValueDecomposition (SVD) is used to smooth the frequency data, and c) synonyms are used to travelvariants of the word-pairs.

DanushkaBollegala*et al.* [11] planned the method which accepted web text wastes to mechanically extracts lexical patterns that describe the relation obscure by the two words in aword-pair and calculates the relational similarity between two word-pairs using a machine learningapproach.

Shuji [10] shape a model to detect defect correction effort based on protracted association rule mining. They defined defect fixing effort as a variable and appropriate association rule mining to treat with such variables. Data used are supported from Japans Ministry of Economy, Trade and Industry (METI).

# International Journal of Advanced Technology in Engineering and Sciencewww.ijates.comVolume No 03, Special Issue No. 01, May 2015ISSN (online): 2348 – 7550

Emad and Walid [13] have developed an approach for predicting re-opened defects through Eclipse projects, their study depend upon some factors such as work habits dimension like: day which issue is closed, the bug report features dimension like: components, the bug correction dimension like: time needed to fix bug.

Weiß et al. predict the "fixing effort" or person-hours spent addressing a defect [7]. They leverage existing bug databases and historical context information. To make their prediction, they use pre-recorded development cost data from the existing bug report databases.

## **III. PROBLEM STATEMENT**

The Software development corporations have to face a lot of problems while preservingphysically all the preservation of the projects, their bugs and their status. This type of problem kinds the whole system an incompetent one and thus making a poor and unorganized working. In order to remove this type of problem, So that the paper is deliberate to develop. The LSI mechanism that resolves the problem contains simply of helpful a very large number of local Data. It also permits the manager of the system to modify the tracking procedure so that unnecessary documentation on the part of the problem solvers does not develop a waste of time.

### **IV. SYSTEM MODEL**

The system model in linear algebra that decomposes a matrix into three factor matrices. SVD is a much more complex approach than other; however, once the original matrix has been moldy, operations on the matrix are rather quick.

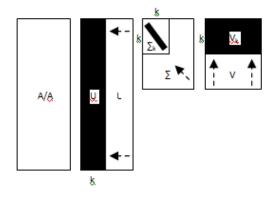


Fig 2: Singular Value Decomposition is a Mathematical

SVD is an extension of Latent Semantic indexing (LSI). LSI is a technique of investigating a group of terms and documents to find relationships between the two [7].

SVD takings as input a matrix of size  $m \ge n$ . This matrix is spoiled into three different matrices:  $U \ge V_T$ . The output of these three matrices is a relationship to the original matrix.

S and  $V_T$  are both orthogonal matrices and  $\sum$  is a diagonal matrix. These three matrices are added identified as:

- S is the right singular vectors
- o Sized:  $m \ge r$
- V<sub>T</sub> is the left singular vectors
- o Sized:  $r \ge r$
- $\sum$  is the singular values
- o Sized:  $n \ge r$

## International Journal of Advanced Technology in Engineering and Science www.ijates.com Volume No 03, Special Issue No. 01, May 2015 ISSN (online): 2348 – 7550

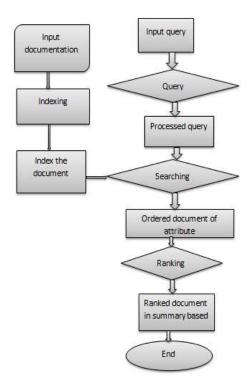
For SVD, "r" is measured to be the rank of the matrix, which is the minimum of the original matrix dimensions. In overall, all matrices necessity is full rank, meaning r is equal to either m or n. In this case, we can accurately rebuild the original matrix given the three decomposed matrices.

SVD has astimulating property that permits for less than full rank of the matrices to approximate the original matrix. For the purposes of this project and LSA, we don't want the original data back (perfect reconstruction of the original matrix), but rather we want underlying relationships in the movie data.

## V. PROPOSED IMPLEMENTATION

To complete Latent Semantic Indexing on a collection of documents, this paper performs the following steps:

- Main in first, change each document in your index into a vector of word occurrences. The number of dimensions your vector occurs in is equivalent to the number of exclusive words in the whole document set. Most document vectors will have great empty patches, some will be quite full. It is optional that common words (e.g., "this", "him", "that", "the") are removed.
- Next, scale each vector so that every term imitates the frequency of its occurrence in context. *I'll column the math for this step when I get home*. Seems he didn't ever get home ;-)
- Next, combine these column vectors into a large *term-document matrix*. Rows represent terms, columns represent documents.
- Execute SingularValueDecomposition on the term-document matrix. This will result in three matrices usually called U, S and V. S is of specific interest, it is a diagonal matrix of singular values for your document system.
- Recombine the terms to form the original matrix (i.e., U \* S' \* V(t) = M' where (t) signifies transpose).
- Break this reduced rank term-document matrix back into column vectors. Assistant these with their corresponding documents.
- Currently you have a Latent Semantic Index.



## International Journal of Advanced Technology in Engineering and Science www.ijates.com Volume No 03, Special Issue No. 01, May 2015 ISSN (online): 2348 – 7550

## 5.1 Indexing Process

A stemming algorithm is a process of languagestandardization, in which the different procedures of a word are summary to a common form, for example,

Connection

Connections

Connective --->connect

Connected

Connecting

It is important to grow that we use stemming with the meaning of improving the concert of LSIsystems. It is not an exercise in etymology or grammar. In fact from an etymological or grammatical viewpoint, a stemming algorithm is accountable to make many mistakes. In adding, stemming algorithms - at least the ones obtainable here - are appropriate to the written, not the spoken, form of the language.

For some of the world's languages, Chinese for example, the concept of stemming is not applicable, but it is certainly meaningful for the many languages of the Indo-European group. In these languages words tend to be constant at the front, and to vary at the end:

ion

ions

connect-ive

ed

ing

The adjustable part is the `ending', or `suffix'. Taking these endings off is called `suffix stripping' or `stemming', and the residual part is called the stem.

## VI. RESULT

The performance evaluation in the thesis is being carried out by using standard Bug tracking of recall and precision for each word&sentence, interpolated average percentage is computed.

Summary/Attributes Based Bug Tracking Classification Using La	tent Semant	ic Indexing				
Non secure Mozilla		Dash Als Dotpos				
				Dependant		Serieto 100 - 7
				Convert (NSSE (WINET) HULL? and Tree Dig Taking Sectors (Industrial Industrial Sectors (Industrial Sectors		10.10340.02
0. Wheen_the335.00THET_PHOVECT lasks_thease dbg TaximgDay_Senses Emilarly/deviloping Taxami 14bit		40 00862392				
D Weeds THEOD COTNET / PICIECT Adds, Theor dug Technology, Jenterics Stellarty derbloater: Second 11 ob		36 5990 1345				
0. Water_14555.00TNET_FROMETTeams_Trees BugTackingSite_SetteronSchlart/dertification/Dataset Dod		12.8077417				
D Made_14555.00 ThET_FRONT New, Team Bug Technolog, Senterior Emistry Methods of Damark That						
0. March, THESH DOTNET, FROMEST Addres, Trease days Technophie, Sectors of Indentification Consult 2004		10.04269415				
0. What _ THEORY DOTNET_PROJECT Make, "Next Bug Factory Dr. Sentence Enterty Verification" Dataset "Data		5.30008849				
		4.79670811				
D Risch_THEOD COTINET_PEORET Anni_Teasi Bug PackingDis_Sense to Similarly Identification Johand Kite						

## Figure 1: Semantic Search Sentence Based Similarity

In Figures 2 and 3, we have presented the results on evaluation of interpolated semantic search for each of the content queries in the database. Figure 3 represents the highlight of sentence for each query with respect to SVD and

LUD at, where we have recorded the maximum average percentageinFigure3.

## International Journal of Advanced Technology in Engineering and Science

#### www.ijates.com

## Volume No 03, Special Issue No. 01, May 2015

ISSN (online): 2348-7550

			Searth	
Search Bugs Here:		Help		
		Index / Re-index	Configure	
Document path			Sedarty-100	T
Provide the product of product of an Equipment of State and State an				
D/Wash_THESE/DOTNET_PHOJECTUITS_LShdugThetergSyx_SummersEmiantyIdentRotation/Dataset/18tet				
D. Wash_THET	19 Indepart		88.04130732	
D-Maid_THE	The Sett Format View Help	20	26.13173546	
D' Misech_THE			22.43876277	п
D/Masch_THE	Summary	1	15.57910364	
D. Wash_THEI			11.025010	
D-Maid_THE	an insecure one	=	8.97189525	
D' M sech, THE			1.44467318	
D Masch_THE			Q 10074075	
D. Masch_THE	processmail bails out when running in secure (taint) mode		-2 62134223	
D-Maid, THE	javascript popup new window causes false warning message with			
D' Misechi, THE			0.92368575	
D Masch_THE			-10.80197758	
D. Masch_THE		*	-13 79000765	
D Mand THES	IS DOTNET_PROJECT-ETS_LShillugTrackingSys_SentenceSoliarty/destrication/Dataset/Soc	10	-15 16830814	
D'Mitsch THES	5-DOTHET PROJECT-JETE LSABueTrackropSys SentenceSmillet/deutification/Dataset/Ltop		-15.41702186	

Figure2: Semantic Search Sentence Based Similarity with Document

		RKSTATE	
Padud	FRERCK	98-5N98-24	0.1.1.1. C. 100 COMMONTATION
Component	SECLATY	SBR-51(588-134	Probably For SER 4 000 (1280)E0(118212
Resolution	99	984.214587.34	
9ats	1.6FE	58F-21 N58F-45	Protesting For NSER-0.00155178542511743

Figure 3: Bug Tracking with Random Attribute

Manage	ement		Here.	
Search Bugs Hore:		Index / Re-index	Configure	
Document path			Snlety/101	
DOM: HELSED DOT	RET_PROACTIVES_IP-IngThenrySe_Sectors/Interpretational Converting		30.00790711	
DOMaich_THESIS-DOT	NET_PROJECT/873_LSNBugTrackingSys_SentenceSinitianty/dentification/Chease(117bit		5.74649653	
D/Maich_THESB\DO	72-Network	a man a	1.90029992	
D-Mover_THESIS-DIO	File Edit Format View Help		1 10280153	
D1M3Hdt_THESISIDO	xpconnect needs a proxy scheme to secure the component manager	- (+ )	0.06451621	
D1Maeds_THESIS DO	specimen needs a proxy seneme to secure the component manager	- <u>1</u>	0.43641809	
D.VM.WOR_THESIS DO			0.32830167	
D-Manh_THESIS-DOT	NET_PROJECT/BTE_LSPBugTrackingSys_SentenceSentertpidentRoaton/Datasen18.tet		0.29973844	
D W such_THESS/DOTNET_PROJECT/BTS_LS/BugTeckingSys_SentenceSimilarlyIdentification/Delawer/List			0.19061608	
D1Wash_THESIS-DOTNET_PROJECT-BT3_LSI-Bug TackingSys_SentenceSellentyIdentScalor/OptimeT.108			0.14435732	
D 1M such_THESIS-DOTNET_PREMIET_UITS_LSP4kgToxsorg5ys_SentenceSelastyIdentification Dataset_Titld			0.02233776	
D1W sect, THESIS DOTNET_PROJECT WTE_LS1AugTackingSus, SectorceSiniasty/dev8cation/Dataset/Sol			-8 04203812	
D. Wieck, THESR DOTNET, PROJECT (BTS J.Sh Bug Taxing Sin, Sentence Switerly Herolication Cassion Edit				
D. Wards, THESIS' DOTNET, PROJECT (BTS _LS) & a Tracking System constrainty development (Dataset 201x)			0.58557598	
D-Wastin_THESIS-007NET_PROJECT/8T3_LSYBugTackingSys_SectorcoSmlatyIdanticulation/Dataset2.htt			-8.58470105	
D-Maath_THESIS-DOTNET_PROJECT/8/11_LSI/BugTackingSys_SentenceSinitiety/dentication/Detailed/Alad				
DOMpach THESE DOT	NET PROJECT/015 LSI-But Tacking/or Sector/SelantidertRotor/Dataset: 11bt		-1.86602586	

Figure 4: Semantic Search Word Based Similarity with Data

## **VII. CONCLUSION**

Current bug tracking systems effectively elicit all of the information needed by user. Without this material developers cannot resolve bugs in a timely detect. We analyze the same bug summary attributes in different

## International Journal of Advanced Technology in Engineering and Science www.ijates.com

## Volume No 03, Special Issue No. 01, May 2015

### ISSN (online): 2348 - 7550

ways. First we analyze the bug summary with the help of LSI Indexing & SVD classification which is simple to use and efficient to other classification algorithm. Natural language processing enables us to implements a more automated and more efficient bug training process. By analyzing same BUG summary with different methods it increase the efficiency of the system. While implementing a range of improvements from these areas may be ideal, bug tracking systems may instead prefer to specialize, thus providing a rich set of choices.

## REFERANCES

- J. Anvik. I, Hiew, and G. C. Murphy, Who should fix this bug? In ICSE'06: Proceedings of the 28<sup>th</sup> International Conference on Software engineering, pages 361-370, 2006.
- P.D. Turney, Measuring semantic similarity by latent relational analysis, in Proc. Of IJCAT'05, pp-1136-1141, (2005).
- [3] S. Artzi. S. Kim and M. D. Emst, Research: Making software failures reproducible by preserving object states. In Ecoop'08: Proceedings of the 22<sup>nd</sup> European Object-Oriented Programming Conference, pages 542-565, 2008. [CrossRef]
- [4] N. Bettenburg, S. Just. A. Schroter. C. Weiss. R. Premraj, and T. Zimmermann. What makes a good bug report? In FSE'08: Proceedings of the 16<sup>th</sup> International Symposium on Foundations of Software Engineering, pages 308-318, November 2008.
- [5] N. Bettenburg. R. Premraj. T. Zimmermann. And S. Kim. Duplicate bug reports considered harmful...really? In ICSM'08: Proceedings of the 24<sup>th</sup> IEEE International Conference on Software Maintenance, pages 337-345, 2008.
- [6] S. Breu, J. Sillito, R. Premraj, and T. Zimmermann. Frequently asked question in bug reports. Technical. Report, University of Calgary. March 2009.
- [7] C. WeiB. R. Premraj. T. Zimmermann, and A Zeller. How Long will it take to fix this bug? In Workshop on Mining Software Repositories, May 2007.
- [8] S. Kim and J. E. James Whitehead. How long did it take to fix bugs? In International workshop on Mining Software Repositories, pages 173-174, 2006.
- [9] P.D. Turney, Similarity of semantic relations, Computational Linguistics, 32(3), 379-416, (2006).
- [10] SHUJI M.AKITO M AND TOMOKO M, Defect data analysis based on extended association rulerning, In Proceeding of International Workshop on Mining Software Repositories (MSR), 2007
- [11] D. Bollegala, Y. Matsuo, and M. Ishizuka. www site the sat: Measuring relational similarity on the web. In Proc. Of ECAI'08, pages 333-337, 2008.
- [12] J.Aranda and G. Venolia. The secret life of bugs: Going past the errors and omissions in software repositories. In ICSE'09: Proceedings of the 31<sup>st</sup> International Conference on Software Engineering(to appear), 2009.
- [13] EMAD S, AKINORI I, WALID I, AHMED H, Predicting reopened bugs: A case study on the eclipse Project. In Proceedings of the 17<sup>th</sup> Working Conference on Reverse Engineering WCRE 2010.