Improvement in Accuracy for Information Retrieval Using Text Mining

Prashant N. Khetade*

*(M.TechCSE.AGPCE,RTMNU,India),

Mr.Shashank Moghe**,Vinod Nayyar***

*(Persistent Systems Pvt. Ltd.India)***
*(faculty M.TechCSE ,AGPCE,RTMNU,India)****

Abstract— Information Retrieval is the most preferred method is used to produce the traceability links between the source code and documentation. This research paper show how Information Retrieval approach is used to establish a traceability links between these two. In the maintenance of the software it is costly and tedious and time consuming process for the software developer to make changes in to the source code .Hence to overcome this problem we develop a system. When we develop software system and documentation for this system. After the delivery of the software if there is any change in the source code which is to be change by the developer is the same time he forget to make a change into the documentation in this case we use above. Information Retrieval to establish traceability links between the source code and the documentation. These IR results into faster efficient traceability links recover the traceability links automatically or semi-automatically and states traceability links, show that this source code is identical for this documentation. Here we propose methods, models, First model is the Vector Space Model (VSM), and Jensen–Shannon Model improves the accuracy. The result of this system has 7.75% of recall values.

Keywords*- Traceability, Requirement, Traceability Links, Information Retrieval, Identifier.*

*****

I. INTRODUCTION

Here in this paper we use system document with the source code used for the traceability by Information Retrieval (IR) method. This IR method is concerned with the retrieval of document from the database. The IR methods can help software maintenance by providing a way to semi-automatically recovering traceability links between the documentation of a system and its source code. For getting this assumption we uses some meaning names for the code items. A lot effort has to be done by the software engineer to improve the explicit connection of documentation and source code, this is achieved using Information Retrieval (IR) techniques for traceability recovery. IR-based methods propose a list of candidate traceability links on the basis of the similarity between the texts contained in the software system. There are several methods which has to be proposed for traceability recovery—e.g., Vector Space Model (VSM), probabilistic model, and Latent Semantic Indexing (LSI). In general, the retrieval accuracy of IR-based traceability recovery methods is assessed through two measures: recall, measuring the percentage of correct links that were found, and precision, measuring the percentage of found links that were correct, the IR methods. The studied IR techniques are the VSM and JSM.

II. ANALYSIS

We perform experiments on four medium-size open source systems, i.e., two additional systems, jEdit and Rhino, in addition to Pooka and SIP, to analyze the impact of Trustrace. Trustrace on four medium-size open-source systems, i.e., jEdit v4.3, Pooka v2.0, Rhino v1.6, and SIP Communicator v1.0-draft, to compare the accuracy of its recovered requirement traceability links with those of state-of-the-art IR techniques. As state-of-the-art IR techniques, we choose the Vector Space Model (VSM), a representative of the algebraic family of techniques, and Jensen-Shannon model (JSM), a representative of the probabilistic family of techniques. We use the IR measures of precision, recall, and the F1 score. We also compare two different weighting techniques: PCA and DynWing. We thus report evidence that Trustrace improves, with statistical significance, the precision and recall of the recovered traceability links. We study the impact of Trustrace on another IR technique, i.e., the Jensen-Shannon similarity model. We perform detailed statistical analyses of the data distribution to select an appropriate statistical test as well as to measure the effect size of Trustrace over JSM and VSM.

Information Retrieval Technique

To build the sets of traceability links, we use the VSM (from the algebraic family of techniques) and JSM (from the probabilistic family of techniques) techniques. Abadi et al. performed experiments using different IR techniques to recover traceability links. Their results show that the Vector Space Model and the Jensen-Shannon model outperform other IR techniques. In addition, these two techniques do not depend on any parameter. Thus, we use both JSM and VSM to recover traceability links and compare their results in isolation with those of Trustrace. These techniques both essentially use term-by-document matrices. Consequently, we choose the well-known TF=IDF measure, for VSM and the normalized term frequency measure for JSM. These two measures and IR techniques are state-of-the-art IR techniques. These techniques are the most useful techniques. In this,we explain both techniques in details.

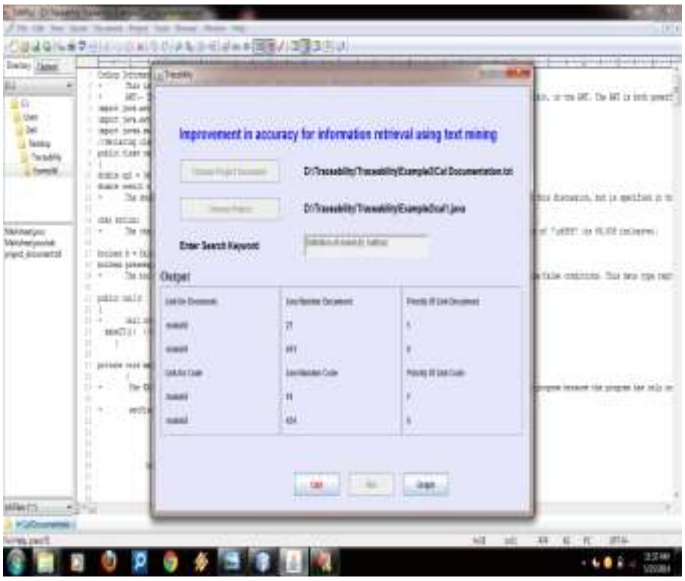
To build the sets of traceability links, we use the VSM (from the algebraic family of techniques) and JSM (from the probabilistic family of techniques) techniques. Abadi et al.[15] performed experiments using different IR techniques to recover traceability links. Their results show that the Vector Space Model and the Jensen-Shannon model outperform other IR techniques. Trustrace uses IR techniques for two different purposes:1) to create the baseline set of traceability links R2C, whose similarity values will be recomputed using Trumo and DynWing using the output of Histrace, and 2) to create the output sets of Histrace, R2CTi;rj;tk .

III. METHODOLOGY USED

In this section we introduce which methodology is used for the development of the system i.e. Project, and how the implementation work goes on. All the implementation strategies and implementation progress is to be explain. NetBeans IDE 7 introduces support for new JDK 7 language features, such as the diamond operator, strings in switch, and multi-catch. When you use these constructs in your code, NetBeans IDE recognizes them, offers correct classes in code completion, correctly highlights errors, and lets you automatically fix old syntax. For the development of the project in a very accurate, easy for understanding of user and for the latest technology end. We use here Java1.7 and NetBeans 7.4 IDE for the development of the project, because the java is the Object oriented programming language as it’s add the various features of OOP’S and the NetBeans is NetBeans IDE 7 is an Oracle sponsored free and open-source Java integrated development environment. Developers from the Java Development Kit (JDK) team have worked closely with developers from the Net Beans team to create a well aligned JDK 7 development experience for Java developers in Net Beans IDE.In the software development of this project we use here Java1.7 and NetBeans 7.4 IDE for the development of the project, because the java is the Object oriented programming language as it’s add the various features of OOP’S and the NetBeans is NetBeans IDE 7 is an Oracle sponsored free and open-source Java integrated development environment.

IV. RESULTS ANALYSIS

In the System design, we now present Trustrace. Trustrace uses software repositories, e.g., mining repositories and bug-tracking systems, as experts to trust more or less some baseline links recovered by an IR technique and, thus, to discard or re-rank the links to improve the precision and recall of the IR-based techniques. Here in this section we done analysis of the result of our system with output and the graph of the system discuss as follows. The project design is to be carried out and implementation work is done, result is obtained for the current develop system which is as follows.



Improvement in accuracy for information retrieval using text mining

D:\Traceability\Traceability\Example3\Cal Documentation.txt

D:\Traceability\Traceability\Example3\cal1.java

Enter Search Keyword

Output

Figure1: System result

. We analyze the result as we compare it here with four systems as jEdit v4.3, Pooka v2.0, Rhino v1.6, and SIP Communicator v1.0-draft, to compare the accuracy of its recovered requirement traceability links. Here the existing system graph is given it show the result up to 6%

We compare the above existing system graph with our developed system graph on these four jEdit v4.3, Pooka v2.0, Rhino v1.6, and SIP. We found the better accuracy and the precision and recall values as with existing system. Our developed system graph showed better result that is up to 80% accuracy in the result. Our develop system graph which is obtain from our develop system on source code and the documentation is as follows.

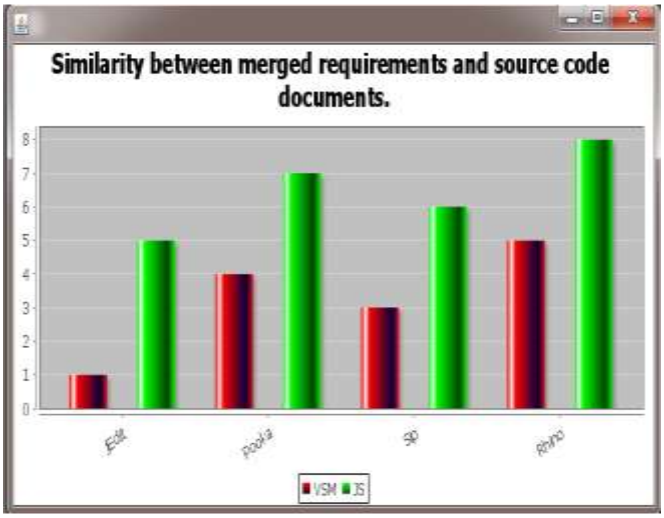


Figure.2: Graph of our develop System.

The existing System graph shows the result 0.6% The results of Trustrace are up to 22.7 percent more precise and have 7.66

percent better recall values than those of the other techniques, on average. Our develop system graph showed result from 2% -6% depending on the various source code and requirement. Experimental results show that, most of the change requests came to the system with an amount of *90%* as expected.

V. CONCLUSIONS

We conclude that IR techniques are useful to recover traceability links between requirements and source code. However, In this thesis work, a software requirements traceability model approach was presented. The proposed model primarily tries to lead the software development teams make efficient and correct impact analysis on the change requests coming to software requirements from both customers and the development team. Moreover, the proposed model makes the development team save a significant amount of change request workload by the underlying software requirements structure which separates and isolates different types of software requirements focusing on the business rules. In future work, we plan to implement more instances of Histrace, in particular using e-mails and threads of discussions in forums.

REFERENCES

[1] N. Ali, Y.-G. Gue´he´neuc, and G. Antoniol, “Trust-Based Requirements Traceability,” Proc. 19th IEEE Int’l Conf. Program Comprehension, S.E. Sim and F. Ricca, eds., pp. 111-120, June 2011.

[2] G. Antoniol, G. Canfora, G. Casazza, A.D. Lucia, and E. Merlo, “Recovering Traceability Links between Code and Documentation,”IEEE Trans. Software Eng., vol. 28, no. 10, pp. 970-983, Oct. 2002.

[3] Marcus and J.I. Maletic, “Recovering Documentation-to-Source-Code Traceability Links Using Latent Semantic Indexing,”Proc. 25th Int’l Conf. Software Eng., pp. 125-135, 2003

[4] J.H. Hayes, A. Dekhtyar, S.K. Sundaram, and S. Howard,“Helping Analysts Trace Requirements: An Objective Look,”Proc. 12th IEEE Int’l Requirements Eng. Conf., pp. 249-259, 2004.

[5] J.I. Maletic and M.L. Collard, “TQL: A Query Language to Support Traceability,” Proc. ICSE Workshop Traceability in Emerging Forms of Software Eng., pp. 16-20, 2009

[6] R. Wu, H. Zhang, S. Kim, and S. Cheung, “Relink: Recovering Links between Bugs and Changes,” Proc. 19th ACM SIGSOFT Symp. and 13th European Conf. Foundations of Software Eng., pp. 15-25, 2011

[7] N. Ali, Y.-G. Gue´he´neuc, and G. Antoniol, “Requirements Traceability for Object Oriented Systems by Partitioning Source Code,” Proc. 18th Working Conf. Reverse Eng., pp. 45-54, Oct. 2011.

[8] M. Eaddy, T. Zimmermann, K.D. Sherwood, V. Garg, G.C.Murphy, N. Nagappan, and A.V. Aho, “Do Crosscutting Concerns Cause Defects?” IEEE Trans. Software Eng., vol. 34, no. 4, pp. 497-515, July/Aug. 2008.

[9] De Lucia, M. Di Penta, R. Oliveto, A. Panichella, and S.Panichella, “Improving IR-Based Traceability Recovery Using Smoothing Filters,” Proc. 19th IEEE Int’l Conf. Program Comprehension,pp. 21-30, June 2011.

[10] Nasir Ali,G Antionol Trustrace-”The mining software Repositories to improve the accuracy of requirement ttaceability links”IEEE transaction on software engg. may 2013

[11] M. Gethers, R. Oliveto, D. Poshyvanyk, and A.D. Lucia, “On Integrating Orthogonal Information Retrieval Methods to Improve Traceability Recovery,” Proc. 27th IEEE Int’l Conf. Software Maintenance, pp. 133-142, Sept. 2011.

[12] N. Ali, Y.-G. Gue´he´neuc, and G. Antoniol, Factors Impacting the Inputs of Traceability Recovery Approaches, A. Zisman, J. Cleland- Huang, and O. Gotel, eds. Springer-Verlag, 2011.

[13] A. Abadi, M. Nisenson, and Y. Simionovici, “A Traceability Technique for Specifications,” Proc. 16th IEEE Int’l Conf. Program Comprehension, pp. 103-112, June 2008.

[14] H. Asuncion, A. Asuncion, and R. Taylor, “Software Traceability with Topic Modeling,” Proc. 32nd ACM/IEEE Int’l Conf. Software Eng., vol. 1, pp. 95-104, 2010.

[15] A. Bachmann, C. Bird, F. Rahman, P. Devanbu, and A. Bernstein, “The Missing Links: Bugs and Bug-Fix Commits,” Proc. 18th ACM SIGSOFT Int’l Symp. Foundations of Software Eng., pp. 97-106, 2010.