International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622
International Conference on Industrial Automation and Computing (ICIAC- 12-13th April 2014)

RESEARCH ARTICLE OPEN ACCESS

Analysis of Improvement in Accuracy for Information Retrieval Using Text Mining

Prashant N. Khetade*, Ms.Sulbha Patil**,Vinod Nayyar***
*(M.TechCSE.AGPCE,RTMNU,India),** (co-ordinator M..Tech. CSE ,AGPCE,RTMNU,India),
*** (faculty M.TechCSE ,AGPCE,RTMNU,India)

ABSTRACT:
The Information Retrieval method is used to produce the traceability links between the source code and documentation. Here we use a IR approach to establish a traceability links between these two. When we develop a 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 as the same time he forget to make a change into the documentation in this case we use above software and the use with Information Retrieval to establish a traceability links between the source code and the documentation. This result to faster efficient traceability links formation and states a traceability links and show that this source code is identical for this documentation. Here we propose a method with three models, First model is the text normalization model, second is Query Extraction Model and the third is Document Classifier with the help of the document classifier we classify the document and the traceability link is form by IR method automatically or semi automatically. The IR method use here are Vector Space Method (VSM)
Keywords - Traceability, Requirement, Traceability Links, Information Retrieval, Identifier.

I. INTRODUCTION

Here in this article for the traceability of the document with the source code used 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 developer 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 artifacts. Such methods are based on the conjecture that two artifacts having high textual similarity share several concepts thus they are good candidates to be traced on each other.

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, LSI, and Latent Dirichlet Allocation (LDA). The first three methods were selected because they are widely used and seem to be the techniques that give the best results. LDA is not as widely used for traceability link recovery though it has been used recently. However, we also experiment such a technique for traceability recovery because LDA is able to capture some aspects missed by other IR methods, such as LSI, when it is used in other occasion. The empirical analysis has been conducted on two software repositories. The studied IR methods have been used to recover traceability links between the use cases and the source code of the two software systems. The results prove that the accuracy of LDA is lower than previously used methods. However, while JS, VSM, and LSI are almost equivalent, LDA is able to capture some information missed by the other exploited IR methods. These considerations suggest that probably LDA can be used as a method to augment canonical methods— e.g., JS, VSM, and LSI—aiming at improving their accuracy.

IR-BASED TRACEABILITY RECOVERY

IR methods index documents and query in a document space by extracting information about the occurrences of terms within them. This information is used to define similarity measures between queries and documents. In the case of traceability recovery, this similarity measure is used to identify that a traceability link might exist between two artifacts, one of which is used as a query. The term extraction is preceded by a text normalization phase. In

particular, in our study we pruned out white spaces and most non-textual tokens (e.g., special symbols, numbers) from the artifact contents. We also used a stop word list to discard common terms (e.g., articles, adverbs) that are not useful to characterize the semantics of the artifact. We also performed a morphological analysis, i.e., stemming, on the extracted terms to remove suffixes of words to extract their stems. The extracted information is stored in a m _ n matrix (called term-by-document matrix), where m is the number of all terms that occur within the artifacts, and n is the number of artifacts in the repository. A generic entry ai;j of this matrix denotes a measure of the weight (i.e., relevance)

II. PROBLEM SOLUTION

In getting the technique term as Information retrieval technique, using this techniques from text summarization in order to create summaries for text based software artifacts. For mixed artifacts, such as, the source code, we need hybrid summarization techniques that combine textual and structural information. Analyzing candidate traceability links is a difficult, time consuming and error prone task, as it usually requires a detailed study of a long list of software artifacts of various kinds. The main issues we are discussing in this paper is how to generate the summaries and then how to evaluate them in the context of the traceability link recovery process. Programmer's process when writing the code is captured by the mnemonics for identifiers. To recover traceability links between code and free text documentation uses the identifiers extracted, and these identifiers is then analyses and get the original query for the source code to match from a class as a query to retrieve the documents relevant to the class. In particular, we apply a vector space IR model to rank the available documents against the class. IR method; ranked list generation; and analysis of candidate links. During the last step a list of candidate links is provided to software engineers for examination.

III. SYSTEM ARCHITECTURE

The developed software system consist of the three portion named as [1] Text Normalization,[2]Query Extraction,[3]Document classifier. The system architecture is shown in below figure.

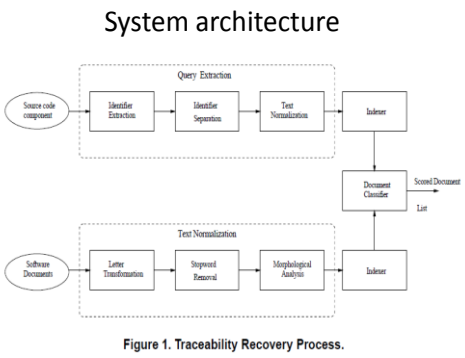


Fig. 1: System Architecture for Traceability Recovery Process

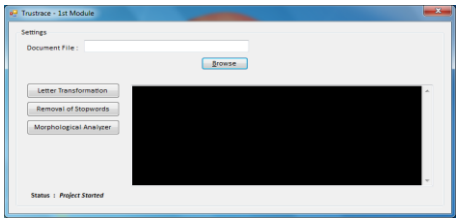
IV. SYSTEM EXPLANATION

In the first module text normalization first we take a text document for a specific software system then perform the following analysis as first we first in letter transformation in the given document we convert all the upper case letter into small, text stop words are removed from the text and last the morphological analysis is done here we remove plural-singular-conjugated bs-infinitives. Here we take C++ source code because it has only 32 keywords (identifier) and develop that system. In the Query Extraction module first we take the source code and the identifier is extracted after the extraction the identifier is separated and the text normalization is to be done. This is shown by the diagrammatically by the figure below. And Once Identifier List has been created for that Code then Vectors are generated for Comparison Purpose. Comparison Results now uses vector Space Model, the Cosine similarity is often used when comparing two documents against each other. It measures the angle between the two vectors. If the value is zero the angle between the two vectors is 90 degrees and they share no terms. If the value is 1 the two vectors are the same except for magnitude. Cosine is used when data is sparse, asymmetric and there is a similarity of lacking

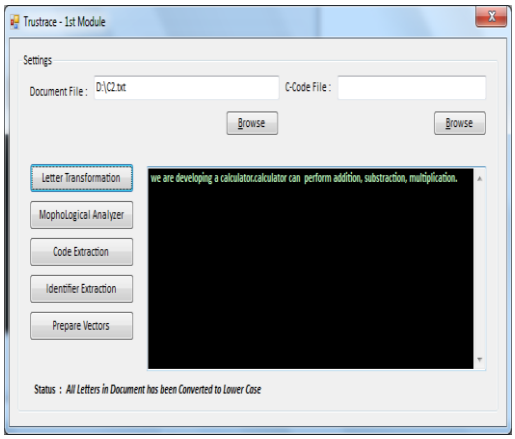
$$cos(\mathbf{x},\mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \cdot \|\mathbf{y}\|}.$$

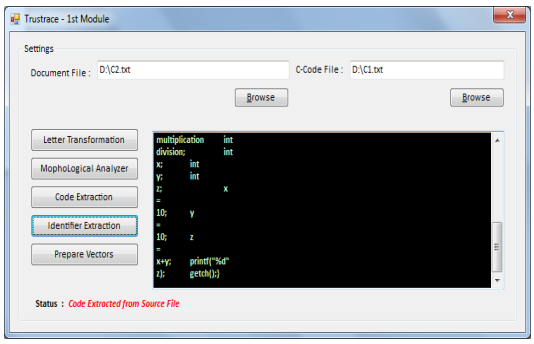
characteristics.

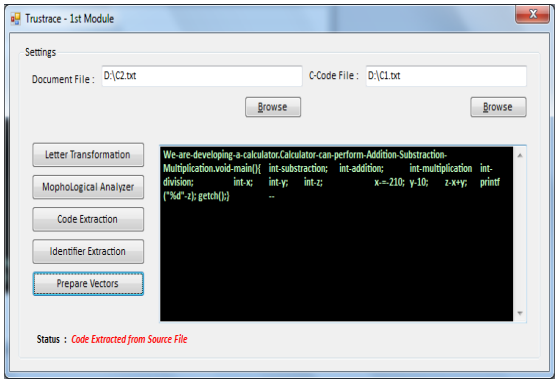
System Module



Browse software code or plain text dialog box.







V. CONCLUSIONS

The Traceability is important for system. The Information Retrieval (IR) approach is most useful for the establishing the traceability links and also for recovering the traceability links. We found that there are many barriers to a company implementing and using traceability links these are to be overcome by using the Information Retrieval approach. It can be further improved to some extent with solutions such as: value based requirements tracing; industry/organisational policies; customised tools; using different approaches and techniques including a semi-automated approach, VSM approach in further implementation and in improving the accuracy. Hence from the above description of the IR approach. It is conclude that the Information retrieval is the better technique in terms of the establishing the traceable links between the Source code and requirement analysis.

REFERENCES

Glean S.Stout "Requirement Traceability" Research paper on Traceability publish in spring cluster 2001.

[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," O.C.Z. Gotel and C.W. Finkelstein, "An Analysis of the Requirements Traceability Problem," Proc. First Int'l Conf. 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] Jaoul paul,Maria Eocob "Requirement Traceability in the Model Driven Approach Applying Model and Transformation Confermence. July 2008

[11] Richard Taokar,Kashif Kamran "Requirement Traceability the State Of Art Systematic Review" in aug 2009

[12] Nasir Ali "Analysis Source Code Structure And Mining Software Repositories to Create Requirement Traceability Links" Montreal in December 2012.

[13] Huzefa Kadgi,Maletic, Bonita Sarif "Mining Software Repositories for Traceable Links" in aug 2009.

[14] Rocco Olivate,Andrea Dee Lucia "On the Equivalenceof Information Retrieval Method for Automated Traceability Link Recovery" in 2010