Indian Journal of Science and Technology, *Vol 9(32), DOI: 10.17485/ijst/2016/v9i32/97209, August 2016*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Ontological based Relevance Abstraction Identification Technique and Evaluation

J. Yesudoss* and A. V. Ramani

Department of Computer Science, Sri Ramakrishna Mission Vidyalaya College of Arts and Science, Coimbatore – 641020, Tamil Nadu, India; jydoss@gmail.com, avvramani@yahoo.com

Abstract

Background/Objectives: To develop ontology based relevance abstraction identification technique for efficient Abstract identification. **Methods/Statistical Analysis:** The abstract terms are extracted from software related documents such as software requirement specifications, compilation report, bug corpus report, library code documents, testing materials and so on. Abstract identification is the process of analysing and identifying the important key words that are present in the requirements document which is essential to understood requirements for better development process. **Findings:** The automated abstraction identification was proposed to extract abstract terms called relevance-based abstraction identification (RAI). RAI-0 and RAI-1 two versions of abstraction identification were proposed. In RAI-1 significance score of term is calculated by assigning variable weights for terms based on the likelihood values where as RAI-0 assign equal weight for all terms. The main issues in RAI is used the lexical similarity which has to improved by using work Ontological based relevance Abstraction Identification (O-RAI) with consideration of conceptual meaning words. This work aims to retrieve the abstract terms by finding the conceptual meaning of every terms present in the requirements document. The O-RAI is implemented by constructing the domain ontology in the automated manner by using the methodology called the episode based ontology construction mechanism.An episode is a partially ordered collection of actions taking place together which is represented as directed acyclic graphs. In episode based ontology construction mechanism, concept attributes and relation among attributes are extracted from episodes, the non-taxonomic relations among attribute also formed based on episodes. **Improvements/Applications:** The significant score of relevant terms from documents is calculated with considering the conceptual of terms which are occurred in the domain ontology. Thus semantic significant score is used to rank the relevance abstract terms.

Keywords: Abstract Identification, Likelihood Values, Ontological based relevance Abstraction Identification, Relevance-based Abstraction Identification

1. Introduction

In recent years, the nature and complexity of software has been changed prominently. To understand about the software engineering it is essential to inspect the quality of software that it builds differently from other things that human beings make. Software engineering is the division of organizations engineering concerned with the improvement of huge and complex software intensive systems. It is mainly focused on the methodology and techniques needed to design and develop complex software systems. For project development there are several kinds of software engineering methods available to complete it on time. These techniques are used to measure the quality of software, bugs and correction.

Software quality is such as correctness, robustness, extendibility, compatible, efficiency, portability, integrity and verifiability. The ability of software should be reusable and changes must be adaptable. Documentation

**Author for correspondence*

is a significant activity in software engineering[1]. Documentation progresses on the quality of a software product. It also plays an important role in software development environment and system maintenance. Many factors contribute to the success of a software project; documentation included. Documentation is one kind of important component which is necessary to provide the information data about the systems.

Various automated methodologies are implemented for improving the confidence level of the final delivered product which can lead to the successful delivery product with high throughput, usability, marketability and ease of support. The worst documentation of the software programming may lead to the errors and causes in the software product development which need to be focused efficiently. Thus the efficiency of the software development will be reduced considerably than the other software development methodologies. It is one of the life activity processes which need to continue till end of the product delivery. It is one kind of tool for making decisions and providing the useful computation scenario. The successful documentation requires well and efficient key points for the software implementation in each and every of the development cycle. By creating the successful documentation one can learn the new successful implementation points that can help for further software development phase.

In Requirements Engineering (RE), the abstraction identification is involved and motivated to provide rich environmental information about the system domain. It performs better than the automatic term recognition (ATR), because the ATR returns the human interpretation of terms. Hence to avoid the problems associated with employing expert human judgment Ricardo Gacitua, Pete Sawyer and Vincenzo Gervasi developed new method named as relevance-based abstraction identification (RAI). This method is efficient in providing usage compare than other methods and human judgment. It is enhanced as RAI-0 and RAI-1 to handle the component words (single and multiwords) effectively.

The main contribution of this work is to identify the abstract terms with the consideration of the conceptual meaning and the location of its presence. This is achieved by implementing the following steps:

- Construct the domain ontology from the textual documents
- Assign the weight values for each word present in the multi terms based on conceptual meaning learned from the domain ontology
- Calculate the significance score and retrieve the top ranked terms as abstract terms.

In this section various previous researches has been discussed in the detailed manner. These researches have been conducted in the area of information retrieval and efficiency. The main approach which is used to retrieve the contents is data smoothing methodologies[2].The data smoothing approaches are classified in two ways. Those are local context analysis vs. global context analysis. By global context, we mean that concepts or related terms are extracted using a knowledge source or a whole collection independently from the input text (document or query). By local context, related terms or concepts are extracted for a given text (document or query) using statistical properties of the sub-collection (top-ranked documents, k nearest concepts, etc.) related to the corresponding text. Finally, we summarize some related works dealing with search context for enhancing document/query representations using either a local context (e.g., a sub-collection, top-ranked documents) or global context (e.g., a whole collection, a single terminology or several terminologies)[3].

Several Domain terminologies have been used by different groups of research in IR, For Example in Bio Medical Domain especially in the context of TREC Genomics[4]. The motivation of TREC Genomics was to support research and development in biomedical IR to drive new experimental research in the area of drug discovery for diseases. Since the commencement of TREC Genomics in 2003 several participants have tried to improve the performance of classical IR approaches by incorporating domain knowledge sources into a conceptual IR model[5]. Generally speaking, conceptual IR model can be viewed a context-sensitive model because conceptual information are extracted within a particular context, e.g., thesaurus, ontology, or related documents, etc. We review in what follows the most termino-ontological resources that have been widely used for indexing biomedical documents.

In order to close the semantic gap between the user's query and documents in the collection, several research works have been focused on applying data smoothing techniques such as document expansion and query expansion on the original document/query. Theoretically, such techniques allow enhancing the semantics of the

document/query by bringing the query closer to the relevant documents in the collection. The semantic information can be detected in a global context (usually from a domain knowledge source or an entire collection) or a local context (usually from a sub collection of related top-ranked documents)[6].

Traceability links between software artifacts are rarely explicit and up-to-date[7,8]. Thus, they have to be identified and maintained during software development and maintenance. Such a task is time-consuming and often is sacrificed under the time pressure of ongoing work[9,10]. The need to provide software engineers with methods and tools supporting traceability recovery has been widely recognized in recent years. Promising results have been achieved by using IR methods for recovering traceability links between different types of artifacts. The idea behind such methods is that most of the requirements documentation is text based or contains textual descriptions, and that programmers use meaningful domain terms to define source code identifiers[11]. Thus, IR-based methods recover traceability links on the basis of the similarity between the texts contained in the software artifacts. The conjecture is that artifacts having a high textual similarity likely share several concepts, so they are good candidates to be traced from one to another.

2. Ontological based Relevance Abstraction Identification

Abstract identification is the process of analyzing and extracting the important key terms which is meant to indicate the concept of the particular document. Abstract terms of particular documents play an important role in helping the software developer for an efficient software development in a specific period and without errors. Here, Identification of the software abstract terms becomes the most important issues which need to be addressed for the well defined software development. In this research work, ontological based relevance abstraction identification is introduced which aims to extract the most conceptual based important terms and terminologies from the documents. The steps followed to implement an abstract identification are given as follows:

- Construct the domain ontology based on conceptual meaning of documents
- Identify the multi terms based relevance
- Assign the weight for each word in multi term with the help of ontology and calculate the significant score of each terms
- Retrieve the abstract terms based on ranking of words

The above are the important steps which are followed to identify the abstract terms of the software. The above steps are discussed detailed in the following sections.

2.1 Construct the Domain Ontology based on Conceptual Meaning of Documents

Domain ontology is a way of representation of documents in specific domain in terms of concepts of various terms present in the document and the interrelationship present between terms. Ontology plays a major role in the real world environment for identifying the semantic meaning and learning the knowledge of particular documents. With the help of ontology, anyone can find the important key words that are present in the documents that can represent the meaning of the entire document. This is enabled by analyzing the interrelation between the different terms that are present in the document.

In this research work, automatic construction of the domain ontology from the requirements documents is done. Domain ontology consists of four layers to represent the relationship between the different terms present in the document. Those layers of domain ontology are namely

→ Domain layer
→ Category layer
→ Class layer
→ Instance layer

Each layer presents the documents conceptual meaning. Domain layer is used to represent the domain name of the documents; category layer represents the various categories that can present under the corresponding domain. Class layer is used to represent the various attributes, operations and their flow in each category. Instance layer also represent the various concepts and their attributes along with their operation. In this work domain ontology is constructed by using the methodology called the episode based ontology construction[12] mechanism. The steps followed to construct the domain ontology are given as follows:

→ Find the most repeated terms using tf-idf term recognition method
→ Cluster the terms which are similar in their concepts
→ Perform episode extraction

→ Construct ontology by defining the attribute operation rules

The above steps are followed to effectively construct the domain ontology.

The domain ontology lead to an finding the most important words present in the multi terms in the effective manner through which weight assignment can be done efficiently. Thus the significant score calculation would provide an accurate result and the ranking can be done in the better way.

The pseudo code for automated ontology construction for the given documents is given as follows:

Algorithm 1: Episode based Ontology Construction

Input: Input documents
Output: Domain Ontology

1. Pre-process the documents. (Removal the stop words which present in the document)
2. Apply stemming (extracting root words from the document's terms)
3. Apply tf-idf to find the most frequently repeated terms present in the document

3.a. Find the tf score

$$tf(t, d) = 0.5+0.5.\frac{f_{t,d}}{\max\{f_{t',d}:t'\in d\}}$$

3.b. Find idf score

$$idf(t, D) = \log\frac{N}{|\{d \in D:t \in d\}|}$$

3.c. Find tf-idf score

$$tf\text{-}idf = tf * idf$$

(Select the term which have high tf-idf score, and omitting the terms which have closed to zero values)

4. Cluster the terms based on their conceptual similarity
5. Extract episodes from the cluster of terms
6. Represent each terms as like follows:

(term, POS, index)

7. Map the terms to the concept clustering
8. Tag the concept name for each terms based on similarity
9. Append corresponding attributes and operation with the corresponding terms
10. Repeat for all terms
11. Return final domain ontology

2.2(i) Identify the Multi Terms based on Relevancy

First the important key terms will be extracted from the documents based on the relevance from the corpus database. The relevance is calculated by identifying the most repeated terms present in the documents. In each and every terms of the documents are ranked based on the relevance score. The relevance score is calculated by identifying the frequency of occurrence of that particular term[13].

Then the log likelihood value is calculated for each and every term which is having highest ranking in order to identify the most important abstract terms[14]. The log likelihood is defined as possibility of term being a most important. It is calculated by identifying the corresponding terms presence in the well defined corpus. The frequency of occurrence of that particular term in the corpus is compared with the actual occurrence in the given document. The ratios of those values are called as log likelihood. Finally the term with most log likelihood is selected as the important abstract in terms of document meaning representation.

The log likelihood is calculated as like follows:[10]

$$LL_w = 2\left(w_d.\ln\frac{w_d}{E_d} + w_c.\ln\frac{w_c}{E_c}\right) \quad (1)$$

Where

w_d→ Number of time presence of word w in source document

w_c→Number of time presence of word w in corpus document

E_d→ Expected value of word in source document

E_c→ Expected value of word in corpus

After calculating the log likelihood values of these terms, terms will be ranked based on them. The higher LL value of the word is said to be term with most confident value. Remaining words are considered as the terms that are not related to document meaning. After extraction of the abstract terms from the documents then the conceptual meaning of them are analyzed in order to make that the extracted words defined the conceptual meaning of the document accurately.

This is done by introducing the ontological based term representation[15] in which conceptual meaning will indicated along with the terms that are extracted which is explained detailed in the following section.

(ii) Calculate Significant Score

In single term representation, relevance score alone is enough to find the most important terms of documents.

However in case of presence of multi terms such as, "This is tajmahal" finding relevance score might lead to accuracy degradation. Each word of multi terms needs to given important for correctly predicting the important terms. Thus the significance score is introduced which will add the relevance score each word in the terms to predict the final result. The signification score for a term t = {w1, w2,.., wn} is calculated using the formula:

$$S_t = \frac{\sum_i LL_{wi}}{l} \quad (2)$$

The above equation doesn't give prioritization for the words present in the terms. This equation gives same priority for all words in the term. For example, in the terms, "This is tajmahal", tajmahal is the most important key word than the other key words. Thus tajmahal need to be given more preference than the other key words. Thus it can be enabled by assigning weight values for each word present in the multi word term. The weighted significant score calculation is done as like follows:

$$S_t = \frac{\sum_i k_i LL_{wi}}{l} \quad (3)$$

The weight k_i is assigned by comparing the words with the domain ontology. The words that represent the concept of particular domain are given more importance than the other words which is differentiated by giving more weight value than other words.

After assigning weight values for each word in the terms, significant score would be calculated which will be appended with the corresponding terms.

2.3 Retrieve the Abstract terms based on Ranking of Words

The terms would be sorted in the descending order based on significant score value that are obtained. Based on this order, the terms would be ranked. The highly ranked terms would be considered as the abstract terms[16].

These abstract terms are then documented alone for further use to the one who will develop the well efficient software without any errors. These abstract terms are used to lead the developers while developing the software by reducing the burden of users. After implementation of this work, experimental evaluations were conducted with various source documents as input. The results and discussion that has been done are discussed in the detailed manner in the proceeding sections.

ALGORITHM: Ontological Based Relevance Abstraction Identification (O-RAI)

Input: a requirements document, Domain Ontology document (*Section 2.1*)
Output: Ontological Based Relevant Abstract Terms

1. Annotate words of documents using POS.
2. Pre-process the documents to filter stop words.
3. Perform stemming on pre-processed documents.
4. Calculate Log Likelihood using equation. (1)

$$LL_w = 2\left(w_d . \ln\frac{w_d}{E_d} + w_c . \ln\frac{w_c}{E_c}\right)$$

5. Find the multi word terms by using syntactic patterns.
6. Find the weight values of the document with the help of ontology k_{oi} for taking each word w_i and in multi word terms (sentence)T_j, to compare the concept with Domain Ontology.
$k_{oi}(w_i, T_j)$ = Score [(Position of Word w_i, in ontology) / Total Count of Concepts in Ontology]

$$k_{oi} = \text{score}\frac{[(\text{position of word in Onto log y}))}{((\text{Total Count of Concepts in Onto log y})]}$$

[Here, score (p(w ,o), (w € c)) represents the position of word in Ontology and the word belongs to the concept, and |(c,o)| represents the total count (absolute value) of concepts in Ontology]
7. Calculate the modified Ontological based significance score using equation. (3)

$$S_t = \frac{\sum_i k_{oi} k_i LL_{wi}}{l}$$

8. Sort the terms based on significant score.
9. Return Ontological relevant abstract terms.

3. Experimental Results

The experimental tests were conducted in terms of various documentation which consists of different types of terms each denotes different meaning. The experimental tests that were conducted for both existing methodology (Relevance based abstraction identification (RAI-1)) and proposed approach ontology based relevance abstraction identification (O-RAI). The comparison is made in terms of the performance metrics called the precision accuracy and recall which are explained detailed in the following sub sections.

The Log Likelihood (LL_w) is needed to be calculated using equation (1). For example, a word in document (hadoop) is considered. $w_d = 4$, $w_c = 6$, $E_d = 1.931$, $E_c = 1.172$, then the LL_w is

$$LL_w = 2\left(4 \ln\left(\frac{4}{1.931}\right) + 6 \ln\left(\frac{6}{1.172}\right)\right)$$

$$LL_w = 2(2.912 + 9.808)$$

$$LL_w = 25.44$$

In total there are 100 terms in the corpus. The abstract terms that are extracted while executing the input source documents are depicted in the figure 1. In this figure, the abstract terms that are retrieved, type of terms and lemma values are shown.

The experimental test values that are obtained are depicted in table 1.

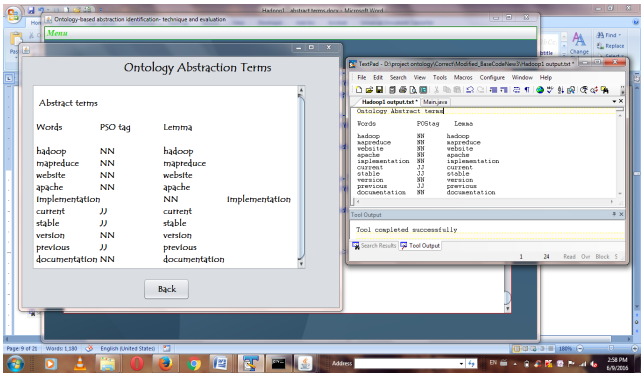

Figure 1. Retrieved Abstract terms with its lemma.

Table 1. Experimental Result Values

Number of Terms	Precision		Recall		F-Measure	
	RAI-1	O-RAI	RAI-1	O-RAI	RAI-1	O-RAI
10	0.9	1.0	0.09	0.1	0.1636	0.1818
20	0.75	0.85	0.15	0.17	0.2499	0.2833
30	0.7	0.8	0.21	0.24	0.3230	0.3692
40	0.725	0.775	0.29	0.31	0.4142	0.448
50	0.62	0.74	0.31	0.37	0.4133	0.4933

3.1 Precision

Precision is defined as the Percentage of correct predicted results from the set of input terms. The precision value should be more in the proposed methodology than the existing approach for the better system performance.

Precision is calculated by using following equation

$$precision = \frac{|\{relevant\ documents\} \cap \{retrieved\ documents\}|}{|\{retrieved\ documents\}|}$$

There are 100 domain terms in the corpus. In RAI-1, the 10 terms namely hadoop, current, stable, version, basics, apache, implementation, mapreduce, website, introduces are considered. When matching, 9 terms namely hadoop, current, stable, version, basics, apache, implementation, mapreduce, website are matched. Hence the precision value is

$$precision = \frac{\{9\} \cap \{10\}}{10} = \frac{9}{10} = 0.9$$

Similarly in O-RAI, the 10 terms are hadoop, mapreduce, website, apache, implementation, current, stable, version, previous, documentation. When matching, all the 10 terms are matched. Hence the precision value is

$$precision = \frac{\{10\} \cap \{10\}}{10} = \frac{10}{10} = 1.0$$

The graphical representation is given in the following figure 2.

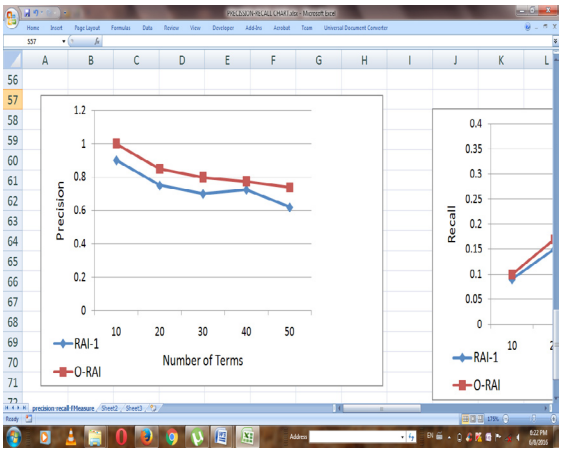

Figure 2. Precision Comparison.

From the above graph it can be proved that the proposed methodology provides better result than the existing approach by selecting correct abstract terms. In this figure x axis plots the number of terms and y axis plots the precision value. Proposed approach improves

62% more than the existing approach in terms of accurate selection of the abstract terms.

3.2 Recall

The recall or true positive rate (TP) is the proportion of positive cases that were correctly identified, as calculated using the equation:

$$recall = \frac{|\{relevant\ documents\} \cap \{retrieved\ documents\}|}{|\{relevant\ documents\}|}$$

As 9 terms namely hadoop, current, stable, version, basics, apache, implementation, MapReduce, website are matched in RAI-1, the recall value when number of terms is 10

$$recall = \frac{\{9\} \cap \{10\}}{100} = \frac{9}{100} = 0.09$$

Similarly, as 10 terms are matched in O-RAI, the recall value is

$$recall = \frac{\{10\} \cap \{10\}}{100} = \frac{10}{100} = 0.1$$

The graphical representation of recall value is plotted in the following figure 3.

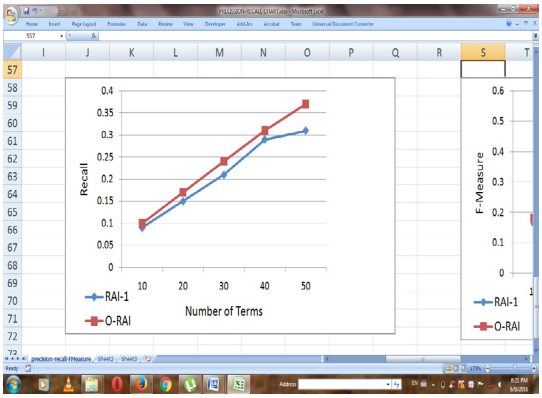

Figure 3. Recall Comparison.

From the above graph it can be proved that the proposed methodology provides better result than the existing. In this figure x axis plots the number of terms and y axis plots the recall value. Proposed approach improves 54% more than the existing approach in terms of accurate selection of the abstract terms.

3.3 F-Measure

The F-Measure computes some average of the information retrieval precision and recall metrics

$$F - measure = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

In RAI-1 when there are 10 terms, 9 terms are matched, the f-measure value is

$$F - measure = 2.\frac{(0.9 * 0.09)}{0.9 + 0.09} = 0.16363$$

Similarly in O-RAI, 10 terms are matched,

$$F - measure = 2.\frac{(1.0 * 0.1)}{1.0 + 0.1} = 0.1818$$

The graphical representation of F-Measure value is given in figure 4.

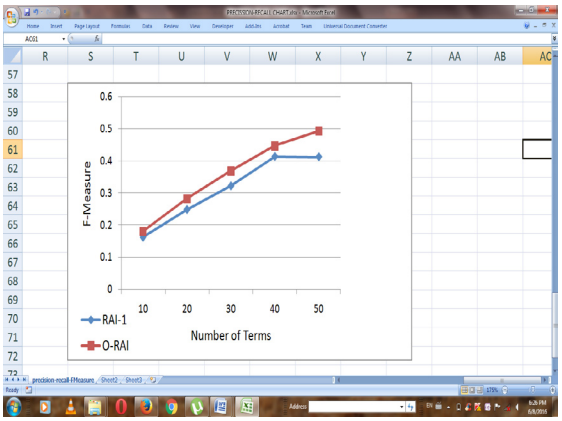

Figure 4. F-Measure Comparison.

From the above graph it can be proved that the proposed methodology provides better result than the existing. In this figure x axis plots the number of terms and y axis plots the F-Measure value. Proposed approach improves 54% more than the existing approach in terms of accurate selection of the abstract terms.

The abstract terms retrieved are listed as follows figure 5.

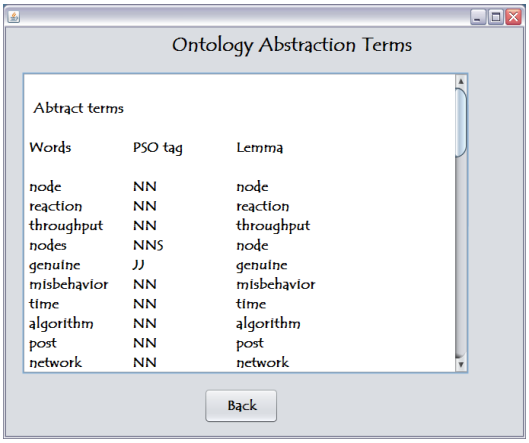

Figure 5. Ontology Abstraction Terms.

4. Conclusion

Abstraction identification is the important role in developing the groundwork steps of software development to the software developer in order to understanding the concept of requirements. In this work, abstract identification is done with the consideration of the conceptual meaning and context sensitive behavior by using Ontology based relevance abstraction term recognition with which abstract terms are extracted efficiently. The conceptual meaning representation is achieved by representing the document in the ontological format. Then the significant score calculations are performed by the help ontology construction in the accurate manner, which leads to effective retrieval of abstract terms for a requirements document. The experimental tests have been conducted and proved that this proposed approach can provide a better result than the existing approach in terms of improved accuracy, precision, recall and F-Measure. Further this research can be utilized with many other methods for the smooth term recognition and in other software areas.

5. References

1. Zimmermann T, Premraj R, Zeller A. Predicting defects for eclipse. In PROMISE'07: ICSE International Workshop on Predictor Models in Software Engineering. 2007; 9–9.
2. Louvan S. Extracting the Main Content from Web Documents. 2009; 303:217–36.
3. Kim S, Whitehead Jr EJ, Zhang Y. Classifying software changes: Clean or buggy? IEEE Transactions on Software Engineering. 2008; 34(2):181–96.
4. Hersh WR, Cohen AM, Roberts PM, Rekapalli HK. TREC 2006 genomics track overview. In TREC. 2006.
5. Kim S, Whitehead Jr EJ, Zhang Y. Classifying software changes: Clean or buggy? IEEE Transactions on Software Engineering. 2008; 34(2):181–96.
6. Kim S, Zimmermann T, Whitehead Jr EJ, Zeller A. Predicting faults from cached history. In Proceedings of the 29th International Conference on Software Engineering. 2007. p. 489–98.
7. De Lucia A, Di Penta M, Oliveto R. Improving source code lexicon via traceability and information retrieval. IEEE Transactions on Software Engineering. 2011; 37(2):205–27.
8. Marcus A, Maletic JI. Recovering documentation-to-source-code traceability links using latent semantic indexing. In Proceedings 25th International Conference on Software Engineering. 2003. p. 125–35.
9. Basili VR, Briand LC, Melo WL. A validation of object-oriented design metrics as quality indicators. IEEE Transactions on Software Engineering. 1996; 22(10):751–61.
10. Gyimothy T, Ferenc R, Siket I. Empirical validation of object-oriented metrics on open source software for fault prediction. IEEE Transactions on Software Engineering. 2005; 31(10):897–910.
11. Takang AA, Grubb PA, Macredie RD. The effects of comments and identifier names on program comprehensibility: an experimental investigation. J Prog Lang. 1996; 4(3):143–67.
12. Lee CS, Kao YF, Kuo YH, Wang MH. Automated ontology construction for unstructured text documents. Data and Knowledge Engineering. 2007; 60(3):547–66.
13. Gacitua R, Sawyer P, Gervasi V. Relevance-based abstraction identification: technique and evaluation. Requirements Engineering. 2011; 16(3):251–65.
14. Yesudoss J, Ramani AV. A survey on abstraction identification techniques in requirement engineering. Karpagam Journal of Computer Science. 2014; 8(3):141–50.
15. Abdul Razak SH, Darleena Eri Z, Abdullah R, Azmi Murad MA. Ontological Model of Virtual Community of Practice (VCoP) Participation: a Case of Research Group Community in Higher Learning Institution. Indian Journal of Science and Technology. 2013; 6(10). Doi:10.17485/ijst/2013/v6i10/38781.
16. Mohankumar P, Vaideeswaran J. Assessment on precision-imprecision essentials in Semantic query processing. Indian Journal of Science and Technology. 2015; 8(13). Doi: 10.17485/ijst/2015/v8i13/55330.