ISSN (online) : 2277-5390
ISSN (Print) : 2277 - 5382

An efficient context sensitive approach for quality of source code and abstract identification

J.Yesudoss [1], M.Pradeep [2]

[1]*Assistant Professor,* [2] *Research Scholar, Master of Philosophy, Department of Computer Science, Sri Ramakrishna Mission Vidyalaya College of Arts and Science, Tamil Nadu.*
[1]jydoss@gmail.com, [2]pradeepmphil10@gmail.com

Abstract

Objectives: The main objective of this research is to achieve the quality of software code and abstract identification with high precision and recall values among diverse software program documents.
Methods: Improved Context-Sensitive Document Recovery (ICSDR) method is used to discover the more accurate abstract terms as well as produce the high quality of software code.
Findings: The proposed method achieves high performance in terms of precision, recall and accuracy.
Application/Improvements: The proposed system is done by using Improved Context-Sensitive Document Recovery (ICSDR) approcah. ICSDR method is used for identification of relevance abstract terms and improves the quality of source code significantly.

Keyword: Abstraction, Context-Sensitive, Information Retrieval and source code identifier quality.

1. Introduction

The software engineering plays a significant role in the knowledge area over the past decade with tremendous growth. Nowadays, the software scenery and difficulty is changed importantly [1]. The issues of software are discovered by using the software engineering methodologies. With the help of novel approaches can efficiently compute the software quality, bugs and correction. In software engineering the software documentation is vital part and it advances the software product quality. Abstraction identification is motivated for obtaining the conceptual information from the specified documents. This is aimed to determine the set of essential models within the problem area that summarized the extent of a visualization structure [2] [3].

Several automated technologies are executed for developing the assurance level of ultimate artifact which can guide to the victorious delivery product along with great throughput, usability, marketability and ease of support. Poor documentation is the reason of many errors and decreases the efficiency in every phase of a software product's development and use. A successful documentation as one that makes information easily accessible, presents a restricted number of user entry points, assists new users learn quickly, make simpler the product and helps cut support costs.

The automatic term recognition (ATR) is introduced in preceding research however it returns the human interpretation of terms. To overcome the drawback of ATR, this scenario is recommended the method named as relevance-based abstraction identification (RAI). This technique is used to handle the single and multiword terms efficiently based on the rank terms. The major work is focused on the identification of more relevant abstract terms from the different documents.

The preceding resaerch has been discussed and suggested several approaches. Kyo Kageura and Bin Umino [4] introduced the method of Automatic Term Recognition (ATR) to extract the relevant terms from corpus. This scenario is suggested the corpus based approaches for computational languages which turn into most popular in the past six years. However it has issues with error rate in the recognition results.

Montreal [5] suggested a collocation extraction approach for developing the robust syntatic analysis. This scenario is targeted on the improvements in customizing machine translation scheme. Hence it has been mined from the given corpus also an important number of source language collocations. However it has problem with prediction of most relevace terms. Daniel Wiechmann [6] recommended the agglomerative hierarchical clustering algorithms to measure the collostruction effectiveness. It is exploited in the process of online conception, for instance, to parse the ambiguous constitution. It is used to predict the relevant abstract terms among verbs and given patterns. But it has issue with overall interpretation of the conclusion.

ISSN (online) : 2277-5390
ISSN (Print) : 2277 - 5382

Pum-Mo-Ryu [7] discussed and produced the concept of domain specific information along with higher accuracy values. This scenario is used to discover the precision of abstract terms with the help of compositional and contextual information. However, it is failed to achieve higher performance due to poor document structure method. Paul Rayson and Roger Garside described the process of key words determination in the corpora which distinguish one corpus from another [8]. This scenario is employed for uncovering the variations among specified corpora and English terms, summary of learner English and text analysis in the software engineering process. However it is not fully automated and hence the significant results might get affected.

In [9] George A. Miller recommended wordnet tool to produce meaningful words, sentences and languages. Wordnet is a collection of English words, nouns, verbs, adjectives, synonyms and attribute relationships. This scenario is sued to provide similarity as well as semantic result for corressponding abstract term. In [10] Leah Goldin and Anthony Finkelstein recommended abstraction based requirements management for facing the challenges of understanding the information in the reuirements engineering. Thus this technique is more suitable for requirement impact study and other requirements association actions. Still it needs an efficient and effective testing method to improve the software.

Another research suggested the method based on document retrieval with superir semantic concepts. Depends on the user queries and specified documents the collection of abstract terms are retrieved by using information retrieval methods. Some other research discussed about traceability issues and it affects the software quality. Traceability links among software products are infrequently precise and up-to-date.

2. Materials and Methods

2.1. Document preprocessing terms

The main aim of the preprocessing is to increase the software quality by reducing the unrelated terms from the corpus. Hence pre-processing is necessary for improve the quality. The document preprocess is performed for token separation and frequency estimation process. Term indexing process is perfromed to calculate the term weight and index process. Document preprocess is to parse the documents into tokens. Stop word removal is used to filter the irrelevant terms. Stemming process is used to take term suffix analysis.

2.2. Term based relevant identification

Abstract identification is described as the process of analyzing and mining the significant key terms from the given documents. This is utilized to identify the concept of specific software. In the software engineering the abstract of specific software is prominent since of the provision of competent software exclusive of errors. Software abstract terms identification is a problem which wants to be addressed for the well defined software growth. In this research scenario, relevance based abstract identification is suggested which targets to mine the necessary terms and methodologies from the software.

The abstract identification is performed through which indication of term based relevance and context sensitive with similarity as well as semantic terms. Initially this scenario is identifying the abstract terms which is depends on the relevance score value. Hence these key terms are mining form the input software code based on the term relevance score value. In the software engineering, the individual terms are ranked by using the relevance score. The terms are sequences of tokens from the given corpus which is identified by the relevance-based abstraction identification (RAI). The relevance score value is computed via recognizing the frquency of occurrence of that speciifc term in the programming.

For each and every term the log probability is measured which consist of the uppermost ranking in order to indicate the more significant abstract terms. It is measured through the indication of corresponding terms existence in the standard corpus. The occurrence of event of that exacting term in the document is evaluated along with the real occurrence in the software programming. The percentages of those values are named as log likelihood. Then the term with most log likelihood is chosen as the significant abstract in terms of document meaning illustration.

ISSN (online) : 2277-5390
ISSN (Print) : 2277 - 5382

The log likelihood is computed as follows:

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

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

The terms have been ranked based on the computation of log likelihood values. The upper LL value of the word is assumed to be term with most confident value. Then the lower and other words are assumed as non relevance to the corpus.

2.3. RAI identification

The multiword and single word terms could be indicated hence in abstraction identification, the terms have to be ranked based on the number of occurrences. A term is a prospective signifier of an essential abstraction in the field of attention. Here the relation among language and concepts is potential, as the huge best part of terms taking place in a corpus will not in fact indicate related abstractions. In RAI, we employ easy syntactic models that conceive multiword terms as frequent arrangements of adjectives and nouns, adverbs and verbs, and prepositions. Each term in the domain text is interpreted along with a part of speech. The group of terms is filtered to eliminate ordinary words improbable to indicate abstractions. The left behind terms are lemmatized to decrease them to their thesaurus form, to disintegrate inflected forms of words to a foundation form or lemma. Then each term is allocated a LL value via using the document-based occurrence profiling method. Syntactic models are used to the content to recognize multiword terms. An important score value is derived for each term through applying the given below formula.

$$S_t = \frac{\sum_i k_i \, LL_{w_i}}{l}$$

S_t is the significance value for a term t = ($w_{1\ldots\ldots}w_l$)

k_i is a weight term

Now the recognized terms are arranged based on their importance score and the resultant list is returned.

2.4. Context sensitive identification

To improve the quality of source code we go for proposed scenario by using context sensitive identifier concept. The method proposed called as Improved Context-Sensitive Document Recovery (ICSDR) to overcome the problem of issues of quality in source code. It checks the abstract with given source code efficiently and effectively by using context sensitive identifier. The concept of each document inputs is analyzed for mining the abstract terms. The method has to compare the specified abstract terms with the corpus and it should be similar as well as semantic. The document which contains terms, sentences and phrases as well as paragraphs. The relevant information is retrieved with the help of selected abstract terms.

In this scenario, indexed corpus and the matching queries are considered as an input for retrieving the abstract terms. Primarily, this method is focused on the removal of redundant queries and symbolizes them in the structured way. Then the preprocessing has been performed for the given corpus in terms of stop word and stemming. The index terms that are equivalent along with input query would attained correspondingly. Based on the ranked score the relevant terms are extracted and hence the quality of result is increased. These kinds of abstract terms are documented for future purpose and it is used to progress the proficient source code excluding of any errors. This is utilized to guide the developers when enhancing the software through decreasing the manual work.

ISSN (online) : 2277-5390
ISSN (Print) : 2277 - 5382

Input:

Context C={$C_1, \ldots \ldots C_n$}

D_{train}={$d_1, \ldots \ldots d$}

T={$w_1, \ldots \ldots w_q$}

(d, AT_i) ←mine(d, AT_i);

M={$(d_1, t_{j,} c_l)$}

Min_a – the value of minimum required number of appearances

Min_i – the minimum value of the threshold parameter of information gain

$L_t^c = \emptyset, \forall\ c, t$

for each c ∈ C

for all document d in D_{train} do

for each term t such that (d, t, c) ∈ M

for each word and phrase w in the sentence of t

if w not belongs to L_t^c

then add w to L_t^c

else

increase the number of appearences of w in L_t^c

define $L^c = \bigcup_t L_t^c$

remove from L^c all words with appearences of less than min_a

remove all words in L^c that appear in T

for each w ∈ L^c compute the information gain

remove from L^c all words where IG (c, w) is less than min_i

return $L^{c1}, \ldots . L^{cn}$

2.5. Performance evaluation

In this section, we consider the performance metrics are such as precision, recall, and accuracy values. In existing scenario, we use RAI method for selecting more relevant abstract terms from the specified corpus. In this scenario, we can formalize the problem of extracting abstractions from a document. In proposed scenario, we proposed this method named as Improved Context-Sensitive Document Recovery (ICSDR). This is used to improve the quality of source code in proposed sysetm. From the experimental result, we can conclude that our proposed scenario is yields higher performance in terms of accuracy, precision and recall rather than existing scenario.

3. Results and Discussion

The existing relevance based abstraction method is used to handle the single and multiword terms from the given corpus. This is assist to a requirements engineer in search to recognize the key abstractions in a new problem area. The Improved Context-Sensitive Document Recovery (ICSDR) is used to improve the software quality by producing efficient abstract documents without errors. An experimental result shows that the proposed method achieves high performance in terms of precision, recall, and accuracy.

ISSN (online) : 2277-5390
ISSN (Print) : 2277 - 5382

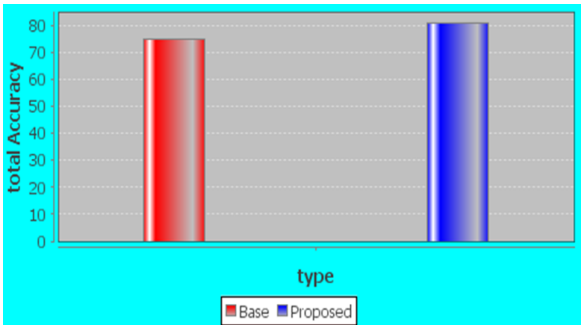
3.1. Accuracy

Accuracy is defined as the degree of generating the experimental output that is matches with the expected output. The accuracy is calculated by using the following equation

$$\text{Accuracy} = \frac{\text{True Positive+False Negative}}{\text{True Positive+True Negative+False Positive+False Negative}}$$

In this graph, x axis is taken for two methods of and y axis is taken for accuracy. From the Figure.1 the proposed scenario shows the highest accuracy rather than existing method. ICSDR provides high quality software by retrieving more abstract terms in proposed method.

Figure 1. Accuracy comparison



3.2. 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

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive+False Positive}}$$

In this graph, x axis is taken for two methods of and y axis is taken for precision. From the Figure.2 the proposed scenario shows the highest precision rather than existing method. ICSDR provides high quality software by retrieving more abstract terms in proposed method.

Figure 2. Precision comparison

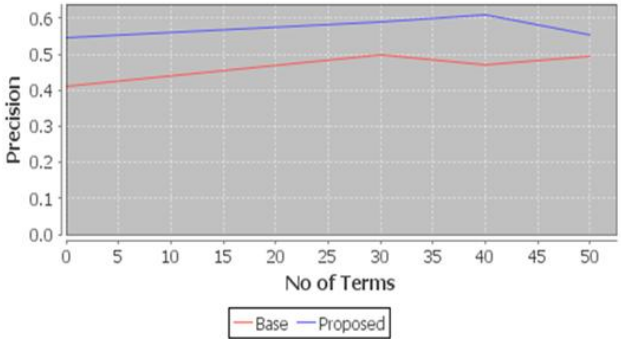
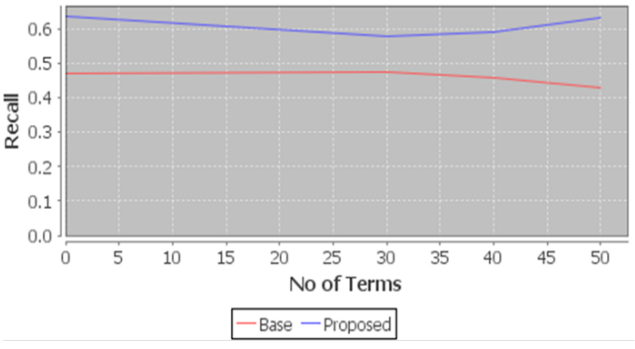


Figure 3. Recall comparison



3.3. Recall

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

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive+True Negative}}$$

In this graph, x axis is taken for two methods of and y axis is taken for recall. From the Figure.3 the proposed scenario shows the highest recall rather than existing method. ICSDR provides high quality software by retrieving more abstract terms in proposed method.

ISSN (online) : 2277-5390
ISSN (Print) : 2277 - 5382

4. Conclusion

In software engineering, the abstract identification is an essential section of source code improvement. This is used to help for analyzing the proper and efficient software requirements. In existing scenario, we used the concept of relevance-based abstraction identification (RAI) for handling the single and multiword terms. Based on the rank score it provides relevance information. However it has issue with producing high quality of source code or software. Hence in proposed scenario we introduced the concept of context sensitive identifier. And the technique name is Improved Context-Sensitive Document Recovery (ICSDR) which handles the abstract terms effectively. It checks the abstract terms with corpus and provides more appropriate result set for corresponding query terms. From the experimental result we can conclude that, proposed scenario yields higher performance rather than existing scenario. The performance is superior in terms of accuracy, precision and recall values. Hence the proposed method is higher accuracy by using context sensitive identifier concept rather than existing scenario.

5. Acknowledgement

We the authors assure you that, this is our own work and also assure you there is no conflict of interest.

6. References

1. Khaled El Emam. Software engineering process, *IEEE transactions*. 2001; ch-9, 1-19.
2. Ricardo Gacitua, Pete Sawyer, Vincenzo Gervasi. Relevance-based abstraction identification: technique and evaluation. *Springer*. 2011; 16(3), 251-265.
3. B.Raghavendra, S.Vasundra. Context awareness for effective software structure quality. *International Journal of Advanced Research in Computer Engineering & Technology* (IJARCET). 2012; 1(7), 221-224.
4. Kageura, Kyo, Bin Umino. Methods of automatic term recognition: a review. *Terminology: International journal of theoretical and applied issues in specialized communication*. 1996; 3(2), 259-290.
5. Orliac, Brigitte, Mike Dillinger. Collocation extraction for machine translation. In Proceedings of Machine Translation Summit IX, 2003, 292-298.
6. Wiechmann, Daniel, On the computation of collostruction strength: Testing measures of association as expressions of lexical bias, *Corpus Linguistics and Linguistic Theory* 4-2. 2008, 253-290.
7. Ryu, Pum-Mo, Determining the specificity of terms using compositional and contextual information, In the proceedings of *ACL 2004 workshop on Student research*, Association for Computational Linguistics, 2004.
8. Rayson, Paul, and Roger Garside, Comparing corpora using frequency profiling, *Proceedings of the workshop on Comparing Corpora*, Association for Computational Linguistics, 2000.
9. George A. Miller, WordNet: a lexical database for English. *Communications of the ACM.* 1995; 38(11), 39-41.
10. Goldin, Leah, Anthony Finkelstein. Abstraction-based requirements management, *Proceedings of the 2006 international workshop on Role of abstraction in software engineering*, ACM, 2006.

The Publication fee is defrayed by Indian Society for Education and Environment (iSee). www.iseeadyar.org

Citation:

J.Yesudoss, M.Pradeep. An efficient context sensitive approach for quality of source code and abstract identification. *Indian Journal of Innovations and Developments.* 2015; 4 (7), November.