JCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Volume-4, Issue-7 E-ISSN: 2347-2693

# Extracting Tasks of Text Files using Dictionary Based Approach for Classification and Indexing

Prachi Rayate[1*], Devendra Singh Thakore[2]

[1] *PG Student, Computer Engineering Department, Bharati Vidyapeeth Deemed University College of Engineering Pune, India*
[2] *H.O.D., Computer Engineering Department, Bharati Vidyapeeth Deemed University College of Engineering Pune, India*

**Available online at: www.ijcseonline.org**

Received:12/Jun/2016 Revised: 20/Jun/2016 Accepted: 16/Jul/2016 Published: 31/Jul/2016

***Abstract*—** In software documentation, product knowledge and software requirement are very important to improve product quality. Reading of whole documentation of large corpus cannot be possible by developers in maintenance stage. They need to receive software documentation entities i.e. (development, designing and testing etc.) in a short period of time. In software documentation an important documents are able to record. There exists a space between information which developer wants and software documentation. This difference can be experimental whenever developers effort to discover the accurate information in the correct form at the exact time. To solve this problem, an approach for extracting relevant task of the documentation under four phases of software entities (i.e. documentation, development, testing and other etc.) is described. The main idea is task extracted from the software documentation, freeing the developer easily get the required data from software documentation with customize portal using Natural Language Processing (NLP) and then the category of task can be generated easily from existing applications. The machine learning approach that is based on supervised learning technique for training dataset in the form of text files based on text mining. Our approach use WordNet library to identify relevant tasks for calculating frequency of each word which allows developers in a piece of software to discover the word usage and also assigning Part-of Speech (POS) to each word. The result shows that task is extracted by calculating how many sentences, tokens and tasks appearing in a document and also shows task is relevant or not. It also reduced a live space between information which developers want and software documentation. This is used to improve the performance of system by taking feedback of developers. The result is identified through customize portal which helps to developers easily get information in a short period of time. The system is 80% precise to extract task by taking feedback of developers in the form of comment.

***Keywords*—** Natural language processing, text mining, part-of-speech tagging, text files, machine learning techniques, WordNet library etc.

## I. Introduction

The system which is use to process sentences in a natural language such as English is called as "Natural language processing". Software requirements and product knowledge are used to improve quality of product. These two are very crucial components to improve quality of product software and product knowledge. The usage of software system rapidly grows increasingly to identify relevant tasks from software documentation within large corpus. To help developers work effectively and minimize software maintenance, there is a serious need for automated support. Data delivery to developers is the main objective of the important document. A lot of technical knowledge and important documents can be able to record in software documentation. There are many forms which are covered in software documentation and these forms are wanted by software developer [1]. To read whole documentation of large systems developers may not get sufficient time and also unable to clear all motivation and purpose of the documentation. This difference can be experimental whenever developers effort to discover the accurate information in the correct form at the exact time.

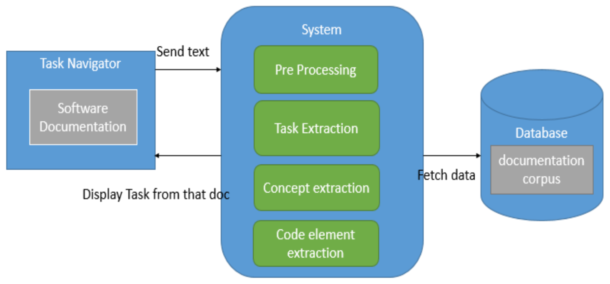

Fig. 1 Task Extraction Process

As the whole document consist of large information the statistical technique, automatically of task structure generally does not prove very useful. To solve this problem, our main idea is to automatically extract tasks from customize portal using Natural Language Processing (NLP).

Many organizations of software development and open source projects try to solve the problem of space between software documentation and the information which developer wants by creating web pages which generated very useful information. As per the user requirements search engines are inadequate and unable to express techniques. The technique gap is reduced by completing the words by search engines which have a software function to do the same and it is presented by high fulfilled user's feedback [4]. For customized search systems the count of previous appeared queries is not too large and also query logs are not available to learn such models [6].

The field of text mining usually deals with texts whose function is the communication of accurate information or opinions, and the motivation for trying to extract information from such text automatically. Text mining" is generally used to denote any system that analyzes large quantities of natural language text and detects lexical or linguistic usage patterns in an attempt to extract useful information.

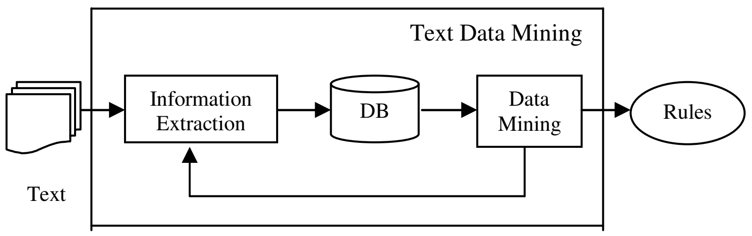

Fig. 2 Overview of IE-based Text Mining Framework

In our system, text mining takes input in the form of text file as a data source. Text files are stored into database to extract task for mining semantically related words. This is useful to structure file from unstructured or raw information. Only text files are use for task extraction in this work. It is easy to extract information from text files because software documentation contains huge amount of data.

The extraction technique combines natural language processing (NLP) techniques, classification methods, text mining and the analysis of syntactical features of the text. Here is need of machine learning techniques for task extraction. This approach requires text mining to summarizes text input and reduce redundancy with NLP. NLP is used to filter each word into a proper way and then whole documentation is analyzed and then categorized into different software documentation phases (i.e. documentation, development and testing etc.). Text mining is the discovery and extraction of interesting, non-trivial knowledge from free or unstructured text. This encompasses everything from information retrieval (i.e., document or web site retrieval) to text classification and clustering, to (somewhat more recently) entity, relation, and event extraction. Natural language processing (NLP), is the attempt to extract a fuller meaning representation from free text. NLP typically makes use of linguistic concepts such as part-of-speech (noun, verb, adjective, etc.) and grammatical structure (either represented as phrases like noun phrase or prepositional phrase, or dependency relations like subject-of or object-of) [2]. The word net dictionary approach is used to find the path of words appears in a document.

The correctness of the processing steps is calculated. Using a standard of sentences and their equivalent the task extraction algorithm is implemented. With the help of 10 software developers and also calculated an evaluation of the tasks extracted from the documentation of customized portal to evaluate whether the extracted tasks are meaningful to developers. The result shows that it reduced a live space between information which developers want and software documentation automatically. And also shows that sentences, tokens and tasks are evaluated. The better results are evaluated by taking feedback from developers in terms of accuracy.

The main contributions of this paper are:

- This work is mainly focus on software documentation to extract some task from large corpus through WordNet library using customize portal.
- Natural language processing is applied to categorize task into different phases and requires machine learning technique for training set of task.
- Part-of-speech (POS) tagging is used to identify task in a document with the help of WordNet library and calculate frequency of each word appearing in a document.

## II. Related Work

In this section, some related works which addresses documentation related issues in task extraction is reviewed. There are many techniques which are already available so this section gives the review of all these techniques. Such extracting issues are very important in software documents. Also compared the propose approach with these available approaches and defines that how the system provides extraction technique to developers.

Treude, Martin P. Robillard, and Barth_el_emy Dagenais described automatically extracting tasks from software documentation is introduced. The tasks which are extracted from documentation that already have been mentioned. Their proposed system is platform independent, which does not require any machine learning languages. One of the important feature of their work is it enables task navigator through the task extracted which is useful for search queries of user interface. They show that software documentation contains sufficient information to extract development tasks and it is use to reduce an information gap between software documentation and developers. This approach is more helpful to developers for identified search results with task of development. Their work based on different techniques for task extraction like natural language processing, statistical techniques, syntactical analysis of text etc [1].

Samir Gupta, Sana Malik proposed a part of speech parsing includes pos tagger and syntactic parser for source code names. They present a POS tagger and syntactic chucker for source code names that takes into account programmers naming conventions to understand the regular, systematic ways a program element is named and identified different grammatical constructions that characterize a large number of program identifiers. Their evaluation results show a significant improvement in accuracy of POS tagging of identifiers, over the current approaches [3].

S. L. Abebe and P. Tonella**,** defines the approaches of natural language parsing to sentences constructed from the terms that appear in program element identifiers. And also shows parsing can be represented as a dependency tree. They automatically extract ontology by mapping linguistic entities (nodes and relations between nodes in the dependency tree) to concepts and relations among concepts. This paper also shows that how queries including ontology concepts reduce the search space, when determining the files relevant to the change request. They have extracted domain concepts from program element names by applying NLP and organizing such concepts into ontology. To validate their approach, they have used the concepts in the ontology to (re-formulate) queries used in concept location [5].

Next, we have focused on query expansion work by Haiduc et al. approach based on the query of properties to improve performance of reformulation strategy by automatically recommends given query. Machine learning consists of training set of data which is trained properly with a sample number of queries and relevant results [11]. Yang et al. used the context addition to reformulated query in which query words are found to extract synonyms, antonyms, abbreviations, and related words [12]. Hill et al. described a similar tool for query expansion. They extracted noun phrases, verb phrases, and prepositional phrases from method and field declarations. Their approach used a hierarchy of phrases and associated a method signature which returns on an initial query [13]. Howard et al. focus on the semantically similar words which are based on software. Their technique mines semantically similar words by leveraging comments and programmer conventions [14]. Again, the main difference to our work is that task extraction suggestions appear after just three typed characters and help the user complete the query rather than reformulate it.

## III. PROBLEM DEFINITION

To propose a highly centralized task extraction approach to keep track of the actual usage of the extraction data in the customize portal to meet the text mining, syntactical analysis, natural language processing and machine learning requirements from the software documentation, which contains all documentation types and also it supports a variety of task extraction methods, categorized each relevant task into different phases of documentation like documentation, development, testing and other etc. and also taking a proper feedback from the developers.

## IV. PROPOSED SYSTEM

A number of issues lead to extract task from software documentation which are related to mismatch information by developers. So to provide an effective mechanism which monitors all the usage of the task on documentation is necessary. In literature survey there are some techniques which already available as discussed but they have some limitations. So to overcome those limitations with the help of applying customize portal framework. All data can be securely extract on documentation is the main purpose of this mechanism. In the existing system, this technique is only applied to particular application but our proposed system apply this framework to text file types which contains all task types, also it also supports a natural language techniques, text mining for text files, machine learning technique using supervised , usage control for extract files. The task is directly extracted from software documentation in existing system but in our idea is to manually extract task, categorized task into different phases i.e. documentation, development, testing and other etc. then after taking a feedback from developers in the form of comment. In next section, the detailed flow of our work is shown.

### *A. Flow Diagram of Proposed System*

In this section, our whole project works modified and then trying to build a customize application that is to give us best result like accuracy in terms of frequency and categorization of task. This approach can be useful to improve the precision of tasks of software documentation. So that it's become easy to reduce a live space between information which a developer wants and software documentations. Basically our work is divided into three phases. Firstly, admin uploads document on the server. Based on that admin add number of tasks which is showed by users. Admin has authority to add number of task into a system. And also specify task description before preprocessing, then choose particular user to assign some task. Whenever admin loads task the machine leaning technique is used. Because data is trained properly and then it is delivered to the number of users using supervised learning. The document can be categorized into different phases and also calculated frequency of each extracted task. Users views documents and extracted task from admin whichever user wants.

Whenever users extracted task at that time admin views all user's data which they have extracted. After preprocessing is done the sentences, tokens, and tasks are calculated in one document. User can comment about extracted task like positive, negative and neutral in the form of feedback using sentiment analysis. User can be able to see all documentation related information like how many sentences, tokens and tasks etc. are there in one file. This dissertation is used supervised algorithm because all data trained properly then it is delivered to the number of users.

*B. Procedure of Task Extraction:*

The data set is stored into the database and take one by one document is nothing but text file. Text mining is applied for discovery of structured data from large corpus. In our case .txt file is supported only for task extraction. Here NLP is applied in such a way that different types of steps are executed. With the help of text mining we summarized data after applying lucene algorithm for indexing of documentation. And each index entries stored frequency of each word in given document. The documentation corpus of a project is also pre-monitored so that to get the extraction of development tasks can be done by text files which kept information as it is. The documentation of each page contains redundant information like summaries, headers and footers which is repeated and also from the files it discarded. During pre-processing the meta-information kept as it is to shows that HTML header represents a paragraph and code is marked up explicitly. By using stand ford NLP toolkit, the resulting files contains pre-tags which is removed and tokens as well as sentences which is also spited into them. Here WordNet library is used to identify each word in a sentence with assigning Part-of–speech (POS) and calculate how many times word occurs in one whole document.

**Task 1:** The sentence is "Testing focuses primarily on the evaluation and the assessment of product quality realized through a number of core practices". After applying natural language processing the sentence is into tokens like "testing", "focuses", "evaluation", "assessment", "product", "quality", "realized", "number", "core", "practices", "involves", "activities". So this type of document comes under testing phase. Each token is assigned Part–of-speech tagging to identify task such as verb and adjective from word net library in task extraction process. In above sentence task is identified like as testing, realize and involve are verbs and quality is adjective. With the help of this task is test quality realize involve after natural language processing is applied.

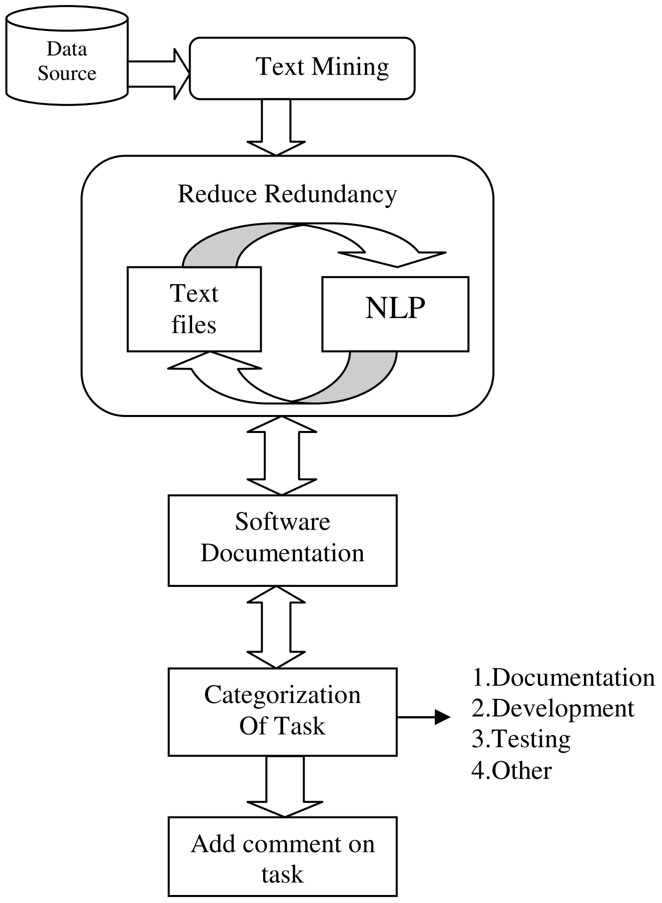

Fig. 3 Proposed System Architecture

To identify each sentence by the StandfordNlp toolkit is the main idea of this work. The whole sentence is tokenized and syntax analyzer is used to check syntax or punctuation occurred in a sentence. In the case to exclude automatically-generate indexes, release notes, and download instructions. Text Mining usually involves the process of structuring the input text deriving patterns in the structured data, and finally the data is evaluated and interpreted. Text mining takes input as a documentation element in .txt file. NLP is applied in such a way that the task is to be filtered and structured with text files based on pre-processing. The extracted tasks are divided into different phases of software engineering (i.e. documentation, development and testing etc.).The tokens and the grammatical dependencies between tokens are stored into files that contain development tasks or relevant concepts can be explicitly excluded from the analysis. In semantic analysis, the Part-of-Speech (POS) tagging is to identify each word in a sentence from WordNet library. Lucene algorithm is used for indexing and searching of words as well as sentences for POS tagging where each document contains index entry. And finally the accuracy of the extracted task to developers gives a proper feedback in the form of comment is calculated.

## *C. Algorithm*

In our proposed system, lucene algorithm for text mining is used.

- *Lucene indexing algorithm*

Lucene is one of the most famous algorithms in text mining for indexing and searching documents which is useful to open source Java library. A term consists of pair of string elements which is the basic unit for searching. This algorithm is especially full text search library written in Java.

1. Determine if the document is new, update in index of not updated in index.
2. If document is already exists then do nothing. If the document is new, create Lucene Document, if it's not updated then delete the old document and create new Lucene Document.
3. Extract the words from the document.
4. Remove the stop words.
5. Apply stemming.
6. Store the created Lucene Document in index.

## *D. Experimental Results*

In our approach, data admin create and upload documentation files on the server. Then admin add relevant task and according to categorize then assign to respective users. Then users search and click on extracted task and give comment like positive or negative about extracted task Admin view all comment about task which has been given by user and also add comment regarding of that task. So whenever next user view task then user will get all information of extracted task. The dissertation work is done which is shown in this section.

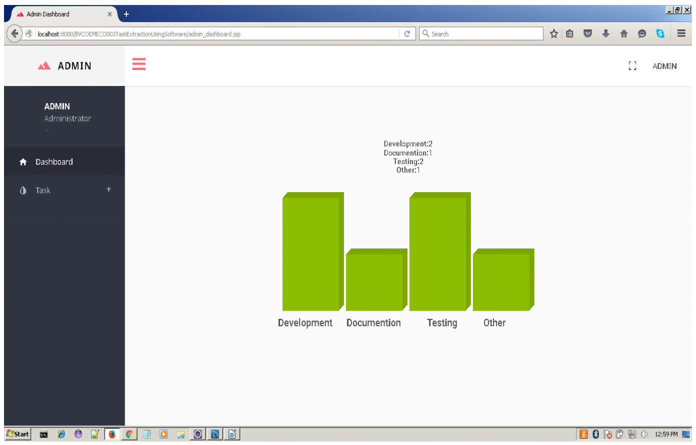

Fig. 4 Categorization of Tasks

In above result, this is dashboard of admin of customize portal. The system is categorized relevant tasks into four phases of software documentation i.e development, documentation, testing and other etc using machine leaning techniques. And also calculated frequency of each task is mapped it in a bar graph and admin is assigned it according to user needs. Admin has authority that which task is given to which user. With the help of this, user will understand how many tasks are there in our system. We can create number of tasks of software documentation in our system.

Table 2 Precision, recall and f-measure statistics

| Document | PRECISION | RECALL | F-MEASURE |
|---|---|---|---|
| D1 | 80 | 2 | 3.902439024 |
| D2 | 81.57894737 | 7 | 12.89364231 |
| D3 | 88.88888889 | 8 | 14.67889908 |
| D4 | 80.76923077 | 5 | 9.417040359 |
| D5 | 80 | 2 | 3.902439024 |
| D6 | 76.92307692 | 6 | 11.13172542 |
| D7 | 83.33333333 | 8 | 14.59854015 |
| D8 | 84.21052632 | 6 | 11.20186698 |
| D9 | 81.08108108 | 7 | 12.88738877 |
| D10 | 87.27272727 | 7 | 12.96046287 |

- *Number of Tasks*

In our system, admin loads number of tasks in customize portal. Such tasks can be extracted by manually and then natural language processing is performed and then uploading tasks on user side. Here the number of tasks is measured. The number of task by changeable the size of documents held by them is measured.

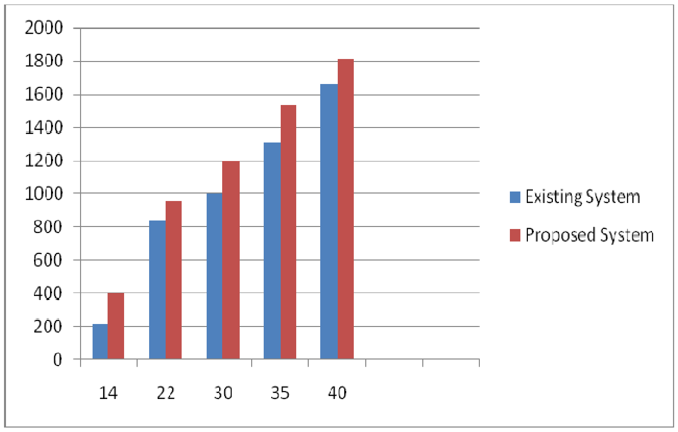

Fig. 5 Number of Tasks in System

In above fig. 5, it shows the number of task i.e the user's document verses the number of created tasks. As per base paper, comparing with existing system, proposed system number of tasks is much smaller than existing system. X-axis shows the number of documents which grows from 0 to 100 and Y-axis shows the values up to 2000 for task number. This number in our proposed system is 14 to 40 for the documents. So as per above results our system give good result in terms of number of tasks. This analysis with values is given in below table 2 which is compared with existing results.

- *Accuracy of Results*

Given the set of documents for each task, the average precision and recall is measured in customize portal. For better performance of system, we randomly selected 10 documents for analysis of result. Fig. shows precision, recall and f-measure of task extraction based on verb and adjective.

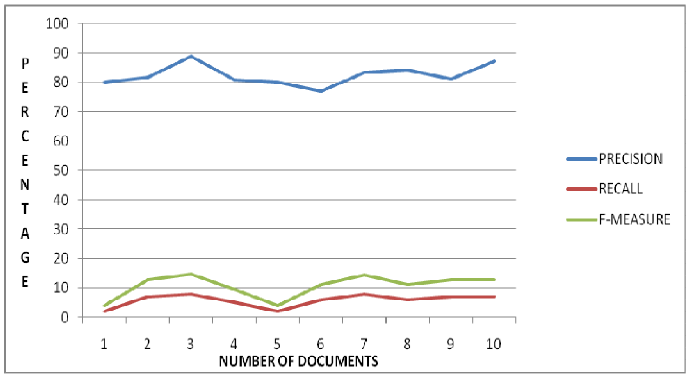

Fig. 6 Precision, Recall and F-measure

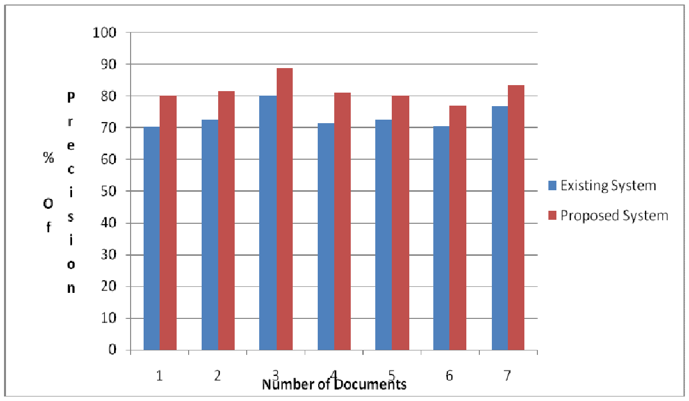

Fig. 6 Comparison of Precision Percentage between Existing System and Proposed System

The above fig. 6 shows that comparison between existing and proposed system is evaluated. This comparison is done with the help of number of documents of extracted tasks. Based on that precision of percentage is calculated which shows that how proposed system is better than existing system. X-axis shows the number of documents which grows from 0 to 10 and Y-axis shows the values up to 100 for percentage of precision. As per the base paper, existing system has smaller precision percentage than proposed system.

## V. CONCLUSION

In this paper, new approach for automatically extracting with natural language processing technique is introduced. This approach allows the developers of the data to not only extract his data but also categorized relevant task into different software phases like documentation, development, testing and other etc. In this concept, NLP is used to get the required data and also used text mining, machine learning, sentiment analysis method and some more algorithms to calculate the results. The result shows that tasks can be extracted from customize portal and also reduced a live space between information which developers want and software documentation also developers can add comment about extracted tasks.

## ACKNOWLEDGMENT

This is to acknowledge and thank all the individuals who played defining role in shaping this paper. Without their constant support, guidance and assistance this paper would not have been completed. Without their Coordination, guidance and reviewing this task could not be completed alone. I avail this opportunity to express my deep sense of gratitude and whole hearted thanks to my guide and head of computer Engineering. Dept. Prof. Dr. D. M. Thakore for giving his valuable guidance, inspiration and encouragement to embark this paper and our Honble principal Dr.Anand Bhalerao at B.V.D.U College of Engineering, Pune (BVDUCOE ).

## REFERENCES

[1] Christoph Treude, Martin P. Robillard, and Barth_el_emy Dagenais ,"Extracting Development Task To Navigate Software Documentation" in Proc, IEEE Soft,Vol.41 No.6,2015,pp,565-581, June **2015**.

[2] S. Gupta, S. Malik, L. Pollock, and K. Vijay-Shanker, "Part-of speech tagging of program identifiers for improved text-based software engineering tools," in Proc. 21st IEEE Int. Conf. Program Comprehension, pp. 3–12,**2013** .

[3] M. Barouni-Ebrahimi and A. A. Ghorbani, "On query completion in web search engines based on query stream mining," in Proc. IEEE/WIC/ACM Int. Conf. Web Intell., pp. 317–320,**2007**.

[4] P. Mika, E. Meij, and H. Zaragoza, "Investigating the semantic gap through query log analysis," in Proc. 8th Int. Semantic Web Conf., pp. 441–455,**2009**.

[5] S.L.Abebe and P.Tonella,"Natural language parsing of program element names for concept extraction," in Proc. 18 th IEEE Int. Conf. Program Comprehension, pp. 156–159,**2010**.

[6] C. Treude and M.-A. Storey, "Effective communication of software development knowledge through community portals," in Proc. 8th Joint Meet. Eur. Soft. Eng. Conf. ACM SIGSOFT Symp. Found. Soft. Eng., pp. 91–101,**2011**.

[7] T. C. Lethbridge, J. Singer, and A. Forward, "How software engineers use documentation: The state of the practice," IEEE Soft., vol. 20, no. 6, pp. 35–39, Nov./Dec. **2003**.

[8] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky, "The Stanford Core NLP natural language processing toolkit," in Proc. 52 nd Annu. Meet. Assoc. Computat. Linguistics: Syst. Demonstrations, pp. 55–60,**2014**.

[9] G. Sridhara, E. Hill, L. Pollock, and K. Vijay-Shanker, "Identifying word relations in software: A comparative study of semantic similarity tools," in Proc. 16th IEEE Int. Conf. Program Comprehension, pp. 123–132, **2008**.

[10] H. Zhong, L. Zhang, T. Xie, and H. Mei, "Inferring resource specifications From natural language API documentation," in Proc. 24th IEEE/ACM Int. Conf. Automated Soft. Eng., pp. 307–318,**2011**.

[11] S. Haiduc, G. Bavota, A. Marcus, R. Oliveto, A. De Lucia, and T. Menzies, "Automatic query reformulations for text retrieval in software engineering," in Proc. 35th Int. Conf. Soft. Eng., pp. 842–851,**2013**.

[12] J. Yang and L. Tan, "Inferring semantically related words from software context," in Proc. 9th Working Conf. Min. Softw. Repositories, pp. 161–170,**2012**.

[13] E. Hill, L. Pollock, and K. Vijay-Shanker, "Automatically capturing source code context of NL-queries for software maintenance and reuse," in Proc. 31st Int. Conf. Soft. Eng., pp. 232–242,**2009**.

[14] M. J. Howard, S. Gupta, L. Pollock, and K. Vijay-Shanker, "Automatically mining software-based, semantically-similar words from comment-code mappings," in Proc. 10th Working Conf. Min. Softw. Repositories, pp. 377–386, **2013**.

[15] James H. Martin, "Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition", by Prentice Hall ,January **2000.**

## AUTHORS PROFILE

**Prachi Rayate** is currently pursuing M. Tech Computer) final year from Department of Computer Engineering from Bharati Vidyapeeth's Deemed University College of engineering Pune, India. Her interests are in Data mining, natural language processing and programming languages.

**Prof. Dr. Devendra Singh Thakore** is working as a head of department in Computer Engineering Department at Bharati Vidyapeeth's Deemed University College of engineering. Pune, Maharashtra, India. He received his M.E. (Computer), M.B.A and P.H.D degrees from various universities. His research interest includes Software Engineering, Cloud Computing, and Data mining.