ISSN No:-2456 –2165

Automatic Development of Task Extraction and Task Recommendation from Software Documents

Atole R.N.[1], Jadhav Komal B.[2], Gaikwad Aaditi U.[3], Lagad Ankita R.[4] , Khartode Seema A.[5]
[2 3 4 5]Student B.E. Computer Engg., SVPM COE, Malegaon, Baramati, Pune

Abstract:-To help developers navigate documentation, we developed a technique for automatically extracting tasks from software documentation by conceptualizing task as a specific programming action that have been described in the documentation. Task extracted from software documentation helps to exits gap between documentation structure and information need of software developer.

Keywords:-Software Documentation, Development Tasks, Navigation, Auto complete, Natural Language Processing.

I. INTRODUCTION

Now a days many software company works on large project there are many possibilities among them are any employee can leave from company or any new employee join to the company. When any employee leave from company and new employee is come to that position then all previous information or code created by previous employee in the form of manual. So it is very difficult to learn all this information to new employee. So company keep Knowledge Transfer period which is only of 2-3 months, so in this period new employee have need to get all details created by previous employee are learn or read in this Knowledge Transfer period. We have to reduce this Knowledge Transfer period by creating all this process online. So we create a search engine when user enter the query then he get all information related to that query his Knowledge Transfer period get reduced by extracting exact information by using search engine in the form of task, concept, code. And main benefit is that reduce the knowledge gap between developers and information need. To help developers to navigate documentation.

II. EXISTING

Manually extracting tasks from software documentation by conceptualizing tasks as specific programming actions that have been described in the documentation.The knowledge needed by software developers is captured in many forms of documentation typically written by different individuals.In existing system many software company keep all records related to work in the form of manual so that when new employee join the company he have to read all the records and study for implementation.

A. Disadvantages of Existing System:

- Manually extract the documents and techniques.
- It is very difficult to find the concept of the documents.
- In Google Search engine we get result according to user view and we get number of result related to that query so it is very difficult to find exact result.
- To find exact result from number of result it is very time consuming process.

III. PROPOSED SYSTEM

Proposed work help to bridge the gap between the information needs of software developers and structure of software documentation. We are developing TASKNAVIGATOR a tool that automatically extracts task descriptions from software documentation. We developed task extraction technique specialized for software documentation and we search in both offline and online mode and additional feature is that user get exact information related to query according to personalization. This application is going to develop for software development industry.

A. Diagram Description

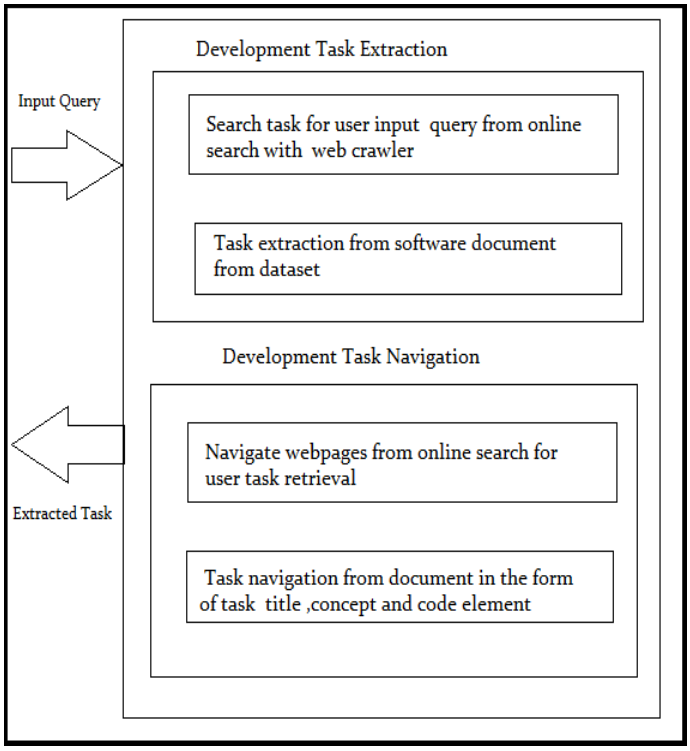

Fig. 1. Block Diagram of Proposed System

Software development task extraction considers software developer as user of the system. User enters search query to extract development task from available project document. This project document contains development task, concept and code element about software project.

B. Working Process

User Input: - Search Query input by user Begin:

- User search query preprocessing by removing stop word from input query grammar processing
- Assigning concept to task (Search query as concept to look up in and as task for search)
- Train Naive Bayes to concept (as query task) classification for relevant document extraction.
- Task classification as cluster (grouping of similar document from dataset available)
- Apply Spy Naive Bayes (Spy Naive Bayes) classification for grouping documents.
- Cluster document task in the form of Task, concept and Code element for search query.

IV. METHODOLOGY

A. Assigning Concepts to Document

This algorithm is implemented for development task extraction from web repository, which shows relevant concepts about search task that is to be extracted.

Example:-If user enter query for adding payment system add this as concept (Task for search) over dataset or web repository. In this process system inputs all available task document from dataset or database schema to concepts for stemming documents names. System shows result of names or content matching web pages or document from dataset to user view. These results are possibly relevant to user query which can be justified by calculating its probability in following algorithm.

B. Training the Naive Bayes Algorithm

This algorithm is implemented for verification of document extracted from web repository or database schema. This produces possibilities of relevant development task from available or total task retrieved. Here, algorithm calculates prior probability and posterior probability of task extraction.

Example:- If user enter query for add payment system will populates result like adding payment, payment solution, payment gateway etc. for checking its relevancy system calculates probability of result occurred in dataset or web repository of dataset.

C. Algorithm for Assigning Task to Cluster

This algorithm is implemented for document classification to its relevant group. In the implementation result of first algorithm, where all task named as relevant concept from that documents of web repository are needed to be grouped. This grouping can be done like positive and negative search document. This algorithm predict development task category for its grouping.

Example:- If first algorithm extract relevant task from web repository, all documents are treated as relevant to user query , so that those documents need to be classified using this algorithm. For instance user query of adding payment first algorithm extract task of add payment ,adding payment, payment gateway ,adding project, new project.so in this algorithm these task are grouped like payment type in same cluster while project is another type or group.

D. The Spy Naive Bayes (Spy NB) Algorithm

This algorithm is used to search relevant task from project repository (Database Schema or dataset) this algorithm is designed to implements task clustering for web pages or document that are extracted. It takes input as output of first algorithm to classify with naive Bayes theorem. This is labeled classification according task category.

V. CONCLUSION

Proposed work help to bridge the gap between theinformation needs of software developers and structure of software documentation. We are developing TASKNAVIGATOR a tool that automatically extracts task descriptions fromsoftware documentation and suggests them in autocomplete inference.

REFERENCES

[1]. ChristphTreude, Martin P. Robillard, and BarthelemyDagenais, “Extracting development tasks to navigate software documentation,” inproc. Ieee trasaction on software engineering, june 2015.
[2]. S. L. Abebe and P. Tonella, “Natural language parsing of program element names for concept extraction,” in Proc. 18th IEEE Int.Conf. Program Comprehension, 2010, pp. 156159.
[3]. G. Antoniol and Y.-G. Gueheneuc, “Feature identification: A novel approach and a case study,” in Proc. 21st IEEE Int. Conf. Softw. Maintenance, 2005, pp. 357366.
[4]. A. Bacchelli, A. Cleve, M. Lanza, and A. Mocci, “Extracting structured data from natural language documents with island parsing,” in Proc. 26th Int. Conf. Automated Soft. Eng., 2011, pp. 476479.
[5]. A. Bacchelli, T. Dal Sasso, M. DAmbros, and M. Lanza, “Content classification of development emails,” in Proc. 34th Int. Conf. Soft. Eng., 2012, pp. 375385.
[6]. 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., 2007, pp. 317320.
[7]. S. Bhatia, D. Majumdar, and P. Mitra, “Query suggestions in the absence of query logs,” in Proc. 34th ACM SIGIR Int. Conf. Res. Develop. Inform. Retrieval, 2011, pp. 795804.