**ABSTRACT OF FYP**

**CODEGO**

**ABSTRACT**

With the Explosive growth of the data on World Wide Web, a large amount of data is available on the Internet. Although today’s software systems are build by the most up-to-date fundamental body which is Source code which is the most reliable data source. It provides a rich and structured source of information, Source code examples are used by developers to implement unfamiliar tasks by learning from existing solutions. To better support developers in finding existing solutions from the large amount of available data of type source code on the internet their should be a design a code search engines to locate and rank code examples relevant to user’s queries Although several systems have been designed to automate this procedure by recommending components on the basis of reusability score metrics that cover the desired functionality which involve ﬁnding reusable software components from online repositories and integrating them to the source code, both to reduce development time and to ensure that the ﬁnal software project is of high quality, but the reusability of these components is usually not assessed by these systems.

Essentially, there is a need for developing code search engine, specifically work for developers and other source-code users.

In this paper, we design the Source code Recommendation System CODEGO to provide a personalized service of code search engine recommendation. CODEGO, a recommendation system for source code components that covers both the functional and the quality aspects of software component reuse. Upon retrieving components, CODEGO provides a ranking schema that involves not only functional matching to the query, but also a reusability score based on conﬁgurable thresholds of source code metrics. The evaluation of CODEGO indicates that it can be eﬀective for recommending reusable source code. We propose a code example search approach that applies a machine learning technique to automatically train a ranking schema on the bases of reusability score metrics of the code. We use the trained ranking schema to rank relevant code examples for new queries at run-time and give recommendations.

A CODEGO first Allow the users or developers for accessing the system by creating an Account, after Registration of the individual user ,user must Login and Provide a Dashboard interface for writing and Searching for Query, user query Tokenized and saved onto the database, system then Crawl the data on the bases of query then provides a ranking schema, which combines a set of ranking features to calculate the relevance between a query and code examples which are extracted from the websites for example(Github,Quora,Stackoverflow).Consequently, the ranking schema places relevant code examples at the top of the result list . We evaluate the ranking performance of our approach using a corpus of over 360,000 code snippets crawled from 500 Github Repositories and provide recommendations to the Developers

The evaluation of CODEGO indicates that it can be eﬀective for recommending source code which not only reduce the software development time but also enhanced the efficiency of the web services to provide better results from large existing of sourcecode data on WWW.

**References**

Bailey P, Thomas P, Hawking D (2007) Does brandname influence percerived search result quality? yahoo!, google, and webkumara, Proceedings of ADCS

Bajracharya S, Ngo T, Linstead E, Rigor P, Dou Y, Baldi P, Lopes C (2006) Sourcerer: A search engine for open source code supporting structure-based search.In: Proceedings of International Conference on Object-Oriented Programming Systems, Systems, Languages, and Applications

Bajracharya SK, Ossher J, Lopes CV (2010) Leveraging usages similarity for effective retrieval of examples in code repositories.In: Proceedings of the eighteenth ACM SIGSOFT international symposium on Foundations of software engineering, pp 157–166

Binkley D, Lawrie D (2014) Learning to rank improves ir in se. In: Proceedings of 2014 IEEE International Conference on Software Maintenance and Evolution BrandtJ,GuoP,LewensteinJ,DontchevaM,KlemmerS(2009