Code search involves finding reusable software components and example code in the increased open source projects. Reusing software components rather than implementing from scratch, makes software development faster, cost effective, and efficient [1]. For this reason, developers spend 19% development time in code search [2]. Usually code search is performed for code reuse, API reference example, auxiliary function, field definition and so on. So the intent of code search can be categorized into two ways - as-is reuse and reference example [3]. When developers use existing code directly, it is known as as-is reuse. On the other hand, when developers use code snippets to gain knowledge about an API or library, this is called reference example. Considering these motivations, current code search engines like google code, krugle, parseweb provides text field to obtain developers query and display code fragments based on that query.

The efficiency and effectiveness of a code search engine depends on the indexing mechanism and query formulation techniques. The reason is that proper indexing and query understanding help retrieving relevant code snippets that satisfy user needs [4]. As the discipline of information retrieval is a mature field [5] so, most of the code search engines employ information retrieval approach for indexing source code [6]. Usually, terms are generated from keywords extracted from source code and index is constructed based on the terms. If two code fragments perform the same task but have different keywords, they will be indexed against different terms. When a user query matches one of the code fragments, search engine should retrieve both code fragments as they perform similar functionality. However, traditional code search engines could not do this because both code fragments contain different keywords. Hence, the recall is reduced for not retrieving such relevant code fragments and the performance of the code search engine is reduced due to low recall [7].

[5] Ramzan, Naeem, et al., eds. *Social Media Retrieval*. Springer Science & Business Media, 2012.