Supporting a Developer's Discovery of Task-Relevant Information

The information that a developer seeks to aid in the completion of a task typically exists across different kinds of software artifacts that include substantial natural language text. For instance, artifacts vary from conversational discussions about bug reports to tutorial descriptions of features in a library. In the artifacts that a developer consults, only some portions of the text will be useful to a developer's task and locating such portions can be time-consuming as the artifacts can include substantial text to peruse organized in different ways. For example, it might be easier to locate information in tutorial artifacts with structured headings whereas artifacts consisting of developer conversations might need to be read in detail.

Given the limited time developers have to spend on any task, researchers have attempted to aid the developers by proposing a range of techniques to automate the identification of relevant text. However, this prior work is generally constrained to one or only a few types of artifacts. Enabling a developer access to artifact-specific approaches is difficult to deploy and support to the multitude of artifact types that is constantly evolving is challenging, if not impractical.

In this dissertation, we propose a set of generalizable techniques to aid developers in locating a portion of text that might be useful for a task. These techniques are based on semantic patterns that arise from the empirical analysis of the text relevant to a task in multiple kinds of artifacts, leading us to propose techniques that incorporate the semantics of words and sentences to identify text likely relevant to a developer's task automatically. We evaluate the proposed techniques assessing the extent to which they identify text that developers deem relevant in different kinds of artifacts associated with Android development tasks. We then investigate how a tool that embeds the most promising semantic-based technique might assist developers while they perform a task. Results show that semantic-based techniques perform equivalently well across multiple artifact types and that a tool that automates the provision of task-relevant text assists developers in effectively completing a software development task.