

Applying Natural Language Processing (NLP) techniques to Software Development:

A limited way of viewing software is as source code alone-treating a program as a sequence of instructions that make it amenable to mathematical techniques. One, however, must not forget that a program, or a software consists of so much more. Developers make use of test cases, documentation, variable names, a version control repository and so many other artifacts. Thus, software analysis tools should deduce more powerful and useful information about the program. The goal of this research paper is to make the case that applying Natural Language Processing techniques to software engineering will foster the creation of tools that allow such useful deductions to be made. The four NLP techniques explored in this project are: Document Similarity, Word Semantics, Parse Trees, and Neural Networks.

The structure of this research project will be a comparative study that examines each of the four NLP techniques listed above, and determine which of them shows the most promise in enhancing the software engineering process. As of this moment, the metric used to gauge the performance of each of these tools will be specification mining-how well each of these techniques can analyze the textual component of a program and produce valid specifications for that program. Today, many programs lack specifications associated with them, making them very difficult to analyze. This motivated the use of specifics as a metric to gauge the effectiveness of the aforementioned NLP techniques. These techniques listed above function by finding bugs and generating code. The different kinds of input these four techniques leverage are error messages, variable names, procedure documentation, and user questions.