Saucygen: Intelligent Codebase Search

Team: Alex Rockliff, James Caddy, Zachary Forman
Industry Partner: Maptek
Supervised by: Christoph Treude

Motivation

Modern codebases are large (Maptek's main repository alone has millions of lines of code), confusing, and often hard for a newcomer to navigate[1]. To make matters worse, engineers typically fail to maintain up to date documentation[2], disdaining even tools like Doxygen that automatically create documentation from comments. Given these circumstances, we created a tool, Saucygen, that extracts understanding from pre-existing code, without reliance on strictly formatted code documentation.

Introducing Saucygen

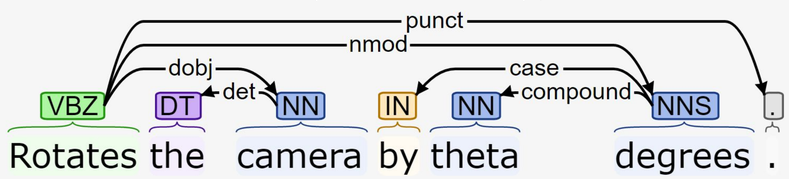
Saucygen is an intelligent codebase search that augments traditional codesearch with understanding extracted from natural language embedded in the codebase.

To give developers an indication of what they can do with a particular element in the code, or how it has been used previously, we find "tasks" associated with those elements. These are described by the authors of the code in notes they might leave within the context of the elements, about how they have used it, or how it is meant to be used. By analysing these notes and using the rules of the English language, we can extract tasks that are useful to developers that are unfamiliar with the code.

Saucygen makes codebases easier to search and use by giving the user the ability to search through these tasks that it finds. The user specifies the action they want to take, or the element they are interested in and receives relevant suggestions.

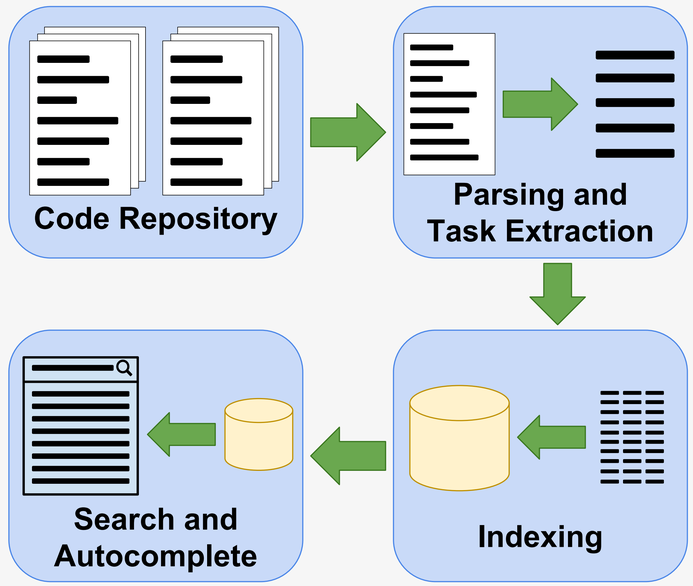
Example Task Extraction

```
// Rotates the camera by theta degrees.
double RotateCamera(double theta);
```



Definitions
double Camera::RotateCamera(double theta);
Comments
Rotates the camera by theta degrees.
Tasks
Rotate the camera.

System Architecture



How Does It Work?

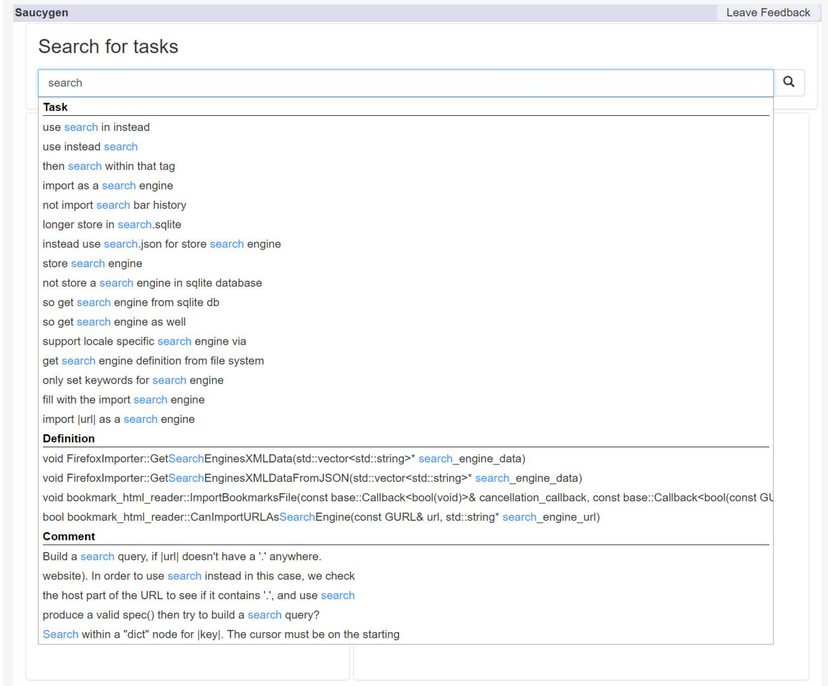
Saucygen consists of several distinct services in a pipeline, as shown in the diagram. Firstly, the codebase is parsed to extract definitions and the associated comments from each file, which are stored in a database along with relevant contextual information. The comments are then analyzed using spaCy[3], a natural language processing library, to extract tasks.

The model that extracts tasks is based on the work presented in Treude *et al's*[4] TaskNav. Input text is analysed, extracting verb pairings with direct objects and/or prepositional phrases to develop the syntax of a task.

A trie-based high performance search index is then constructed from the tasks, comments, and definitions, retaining links to the contextual information in the database.

This search index is then served using a Flask backed web server so developers can search with the assistance of autocomplete. After performing a search, content relevant to the selected result will be displayed in the results pane, along with a link to the full source file.

User Interface



Future Work

There are several plausible paths for extending and improving Saucygen. One issue is that search requires the developer to know the terminology the codebase uses. Searching "viewpoint rotation" when "camera facing" is the term used will lead to no results. To alleviate this, we propose the use of word vector schemes to allow for semantic lookup of terms - i.e. searching for the intent of a phrase as opposed to the words themselves.

A secondary issue is that spaCy, the natural language processing library we have used, occasionally fails to correctly parse text. Google's Syntaxnet is significantly more accurate, and utilising it could offer greatly increased accuracy at the cost of index time.

References

[1] Steinmacher, Igor, et al. "The hard life of open source software project newcomers." Proceedings of the 7th International Workshop on Cooperative and Human Aspects of Software Engineering. ACM, 2014.
[2] Spinellis, Diomidis. "Code documentation." IEEE software 27.4 (2010): 18-19.
[3] SpaCy [https://github.com/explosion/spaCy]
[4] Treude, Christoph, et al. "TaskNav: Task-based navigation of software documentation." Proceedings of the 37th International Conference on Software Engineering-Volume 2. IEEE Press, 2015.

CRICOS Provider Number 00123M