

A Study of Ranking Schemes in Internet-Scale Code Search

Sushil Bajracharya*, Trung Ngo*, Erik Linstead†, Paul Rigor†, Yimeng Dou†,
Pierre Baldi†, Cristina Lopes*

*Institute for Software Research †Institute for Genomics and Bioinformatics

{sbajrach, trungcn, elinstea, prigor, ydou, pfbaldi, lopes}@ics.uci.edu

Institute for Software Research
University of California, Irvine
Irvine, CA 92697-3423
ISR Technical Report # UCI-ISR-07-08
November 2007

Abstract. The large availability of source code on the Internet is enabling the emergence of specialized search engines that retrieve source code in response to a query. The ability to perform search at this scale amplifies some of the problems that also exist when search is performed at single-project level. Specifically, the number of hits can be several orders of magnitude higher, and the variety of conventions much broader.

Finding information is only the first step of a search engine. In the case of source code, a method as simple as ‘grep’ will yield results. The second, and more difficult, step is to present the results using some measure of relevance with respect to the terms being searched.

We present an assessment of 4 heuristics for ranking code search results. This assessment was performed using Sourcerer, a search engine for open source code that extracts fine-grained structural information from the code. Results are reported involving 1,555 open source Java projects, corresponding to 254 thousand classes and 17 million LOCs. Of the schemes compared, the scheme that produced the best search results was one consisting of a combination of (a) the standard TF-IDF technique over Fully Qualified Names (FQNs) of code entities, with (b) a ‘boosting’ factor for terms found towards the right-most handside of FQNs, and (c) a composition with a graph-rank algorithm that identifies popular classes.