Reviewer: 3

Public Comments (these will be made available to the author)

The authors cherrypicked from among the criticisms and suggestions I made in

the initial review. Of those they ignored, I list the most important below.

The authors' study of similarity metrics is technically incorrect: it rests on

a unit error. It compares the authors' term granular (here term means

subtoken, extracted via stemming based on camel case or underscores) against

two symbol (or character) granular metrics. Obviously, a term is not a symbol;

it is a contiguous, unbounded sequence of symbols. Thus, they have given their

measure an unfair advantage, since it operates over units that are arbitrarily

larger than symbols.

I pointed this out in the first review and even suggested a means of rectifying

the problem, by constructing a alphabet of "symbols" from their terms for use

by Levenshtein and Jaro-Winkler. The authors sought to address this merely by

observing the granularity mismatch in the introduction. This is inadequate.

It is also strange that at least one of the authors also published a paper at

OOPSLA'17 that uses Needleman-Wunsch, not the current draft's bespoke

similarity measure. This paper does not consider Needleman-Wunsch. Therefore,

both papers implicitly disagree about which distance measure to use. The

authors should have resolved this discrepancy.

Our comments: the comments above are all about our metric. So we can emphasize in the new version that the only purpose of section 3.8 is to validate whether the conclusions drawn with our name-based analysis hold when different similarity metrics are employed, and which metric is better than others.

In my initial review, I asked the authors: "What is the core scientific

contribution of this work over the conference paper [at FSE'16]?". I did not

ask this question idly. For this reviewer, adding four research questions

whose answers require only descriptive statistics and increasing the size of

the corpus is insufficient. Here again, the authors ignored my question.

Our Comments:

Concerning the research questions, in my initial review, I observed "The

presentation of the research questions is ad hoc. What theoretical

considerations inform and unite them?" Here again, the authors ignored

me.

"Statically resolving methods calls is an approximation that may miss

overridden methods" is false. If it were true, compilers could not build

dispatch tables. It is possible to soundly over-approximate which methods

\*may\* be called from a particular call site, and therefore not \*miss\* any.Knowing which of the methods that may be called that will be called is the hard

problem. Section 3.4 that the authors added to address my concern here needs

work. The restriction to overridden methods in the quote above and in the

title of 3.4 is confusing, because the fundamental problem is polymorphism.

The example in Fig. 9 is unneeded; the problem is unclear use of terminology.

Static analysis does not "resolve method invocations incorrectly"; as already

stated, it soundly determines which methods \*may\* be called. This use of

"incorrect" is \*incorrect\*. Further, it is not at all clear that computing the

lexical similarity of arguments to formals computed across all the functions

that may be called from a particular call site would not, in fact, be

profitable.

Our comments: we will change the title of 3.4 to “Impact of Polymorphism on Results of Parameter Names”, and modify the statement about static analysis and overridden methods.

The authors ignored my request to define "named variables". They cannot mean

temporaries because these would fall under their non-variable category. I can

only conclude that "named variables" is incoherent, because there is no such

thing as an unnamed variable.

Our comments: We change all “named variables” to “variables”.

They ignored my observation that Java and C are related languages wrt to

function call syntax.

Our comments: It is a little difficult for us to change to another language now. It takes time to do experiments again.

The work rests on the assumption that lexical similarity can proxy semantic

similarity. As I pointed in my initial review, the authors ignored the

challenge to this assumption posed by synonyms, conventions like prefixing, and

abbreviations, universal names in library calls vs the domain specific names

applications tend to use. In effort to address this point, the authors added a

case study, however, they ignored my suggestion that they make this assumption

explicit (with a forward pointer to this case study) in the introduction.

Thus, this crucial assumption remains buried deep in the paper. Moreover, this

case study was rushed. It is poorly written, with many tense errors. What

procedure did the raters use to make their assessment? For instance, did they

examine the code or just discuss the names? Finally, discussion to consensus

is notoriously unreliable. The authors should have computed interrater

agreement.

Our comments: We have pointed out the assumption in the 4 paragraph of Section 1, and describe the process of our case studyin more details (in the last but one paragraph of Section3.1) .

Another comment I made in the previous review that the authors ignored follows:

"Ignoring the base expression of a call, the object name of a field access, and

the handling of this are important experimental design decisions that the

authors make without justification. The authors should experimentally justify

these decisions; they should report the effect different combinations of these

decisions has on their results."

This statement betrays a misunderstanding of basic statistics: "The p-values

for all of the coefficients turn out to be zero, suggesting that it is

impossible [sic] to see such observations if the similarity is not related to

the length of names."