

Assignment 1

Huan Sun

Apart from those already covered in the assignment, other difficulties I often encounter during software development are:

- When I pose a query, too much information can be found on the Web, and it takes a lot of time to finally get what I really want. There are two kinds of queries I often pose in a search engine during software development: (1) Queries on how to do something, e.g., “How to traverse a dictionary in C#” and “How to make/implement/install X ”; (2) Queries for debugging, e.g., “assertion failure in Y ” and a Java exception. Some queries are easily to resolve by looking at the first few pages returned by Google, say “How to traverse a dictionary in C#”. But very often, especially when dealing with some unexpected exceptions and failures, I feel hard to quickly get useful information to solve my problem. For example, discussion forums such as StackOverflow are usually what one resorts to during this process. For one query, one might get many responses, each of which might also have their own comments and follow-ups. I think in these cases, a summarization tool in discussion forums which can present succinct answers to a user query might be of great help. A **1ood summarization tool** might need to decide what information the user is asking, differentiate the key aspects in responses such as “Why the user is getting this error” (reason) and “What the user needs to do” (action) and list them separately. Therefore, I think more research efforts could be devoted to developing such tools.
- It is hard to read and understand code snippets written by other people, especially those not properly commented. In a large component, while it is easy to understand each instruction (e.g., by checking API documentation or usage of basic operators), one can lose the big picture on what the entire code chunk is doing. It might be too demanding or time consuming for programmers to write all the comments and annotations manually. I am thinking automatically annotating a code snippet might be useful: Based on the description for each instruction in the code snippet, generate a **2igh-level description** about what the code is doing. More exciting would be if a tool can automatically **3egment a long code snippet** based on the functions (or, what they try to achieve) and then annotate each segmentation. Such annotations could also help search for such snippets in the future. I need to investigate the state-of-art research results in this aspect, though.
- Proactively detecting some defects in the code can be difficult. Without any exceptions or error reports, it is hard even to realize there is something wrong in the code before sufficient testing with

Summary of Comments on sun.pdf

Page: 1

1 Number: 1 Author: mernst Subject: Highlight Date: 1/12/2016 6:13:28 AM

This seems like a very interesting (and challenging) topic to me. It might be helpful for you to look at a dozen or so examples and try to summarize them yourself. What information do you use, how do you know to use it, and what does the summary look like? This will help to make the problem more concrete. Equally importantly, it will help you to scope this problem to a reasonable size rather than trying to solve the entire summarization problem. Yours might think about what summarization has been done in the past. That is, you should read the related work.

1 Number: 2 Author: mernst Subject: Highlight Date: 1/12/2016 6:16:05 AM


This a good problem, but can you be more specific about it? What certain information you looking for? Can you characterize the sorts of documentation that you would find useful or not useful? Different types of documentation are likely to be useful in different situations. One nice thing about such a tool is that it could generate documentation based on the user's current need. Experienced users and inexperienced users might meet need different types of documentation, and users who are performing different tasks might be different types of documentation. Rather than having the code cluttered with all of that different documentation (actually, it's unrealistic that all of it would ever get written), it could be generated on the fly and the user would only see what is most relevant to them.

I don't suggest that you try to solve all problems in the space; however, thinking about some specific task in solving that one would be a worthwhile step.

1 Number: 3 Author: mernst Subject: Highlight Date: 1/12/2016 6:17:21 AM

Okay, this is a concrete, modest, but still interesting instantiation of your problem. It's the sort of thing was looking for in my previous comment. (But I hadn't read to here yet.)

many inputs and desired outputs. I am wondering whether there are some analysis tools that can automatically detect “abnormal” usages of variables or give some warnings when using a method. For example, there is some discussion about the error related to “Object.Equals” in StackOverflow (<http://stackoverflow.com/questions/2219047/net-dictionaries-have-same-keys-and-values-but-arent-equal/2219052/#2219052>). Actually in the method usage covered by MSDN tutorials, there are remarks specially dealt with the error (<https://msdn.microsoft.com/en-us/library/bsc2ak47.aspx>). With that said, if a tool can automatically recognize there are pitfalls in using a certain method (by referring to a tutorial or documentation of that method), it might be good to give some warning to the programmers. This might save programmers a lot of time, compared with tracing down what might be going wrong later.

 Number: 1 Author: mernst Subject: Highlight Date: 1/12/2016 6:22:15 AM
Or, recognize when the specific use is potentially problematic.

One common way of doing this is to recognize (that is statistically generate) common usages. For example, if 99% of the time a call to read is preceded by a call to open, then a heuristic warning can be issued whenever read is called without first calling open. This idea of finding common patterns and warning when they are not followed is quite general and you could imagine applying it to a new problem.

Francis, in your example, you could see that usually when a dictionary is defined, it overrides equals. Whenever dictionaries define that does not override equals, the programmer could be warned. There exist tools that have such rules hardcoded but would be nice to automatically generate the rules. These rules could be plugged into one of those existing tools, or could be in a tool of its own.