Presentation - XSnippet: Mining for Sample Code

Tao Xia

November 8, 2006

Motivation

Programming Context

- Parent Context
  - CP(m) is the parent context of method m.
  - CP(m) includes superclass and interfaces implemented by the class which m belongs to.
  - All methods in one class share the same parent context.
- Type Context
  - CT(m) is the type context of method m.
  - CT(m) includes sets of types for inherited fields, local fields, and lexically visible types within the scope of m

Snippets Queries

- Generalized instantiation query
  - – Input a type t, returns the set of all snippets that instantiate the type t

$$\mathcal{I}Q_G(t_q) = \forall s \in \mathcal{S} : scontainst_q instantiation$$

Snippets Queries

- Type based instantiation query
  - Input a type t and the type context CT(m), returns the set of all snippets which t is instantiated from some type tc from CT(m)

$$\mathcal{I}Q_{\mathcal{T}}(t_q, \mathcal{CT}(\mathtt{m})) = \exists s \in \mathcal{I}Q_{\mathcal{G}}(t_q) : \mathcal{T}(s) \cap \mathcal{CT}(\mathtt{m})$$

Snippets Queries

- Parent-based instantiation query
  - Input a type t and the parent context CP(m), returns the set of all snippets which the class of the snippet and the class of m either inherit from the same class or implement the same interfaces

$$\mathcal{I}Q_P(t_q, \mathcal{CP}(\mathtt{m})) = \exists s \in \mathcal{I}Q_G(t_q) : \mathcal{CP}(s) \cap \mathcal{CP}(\mathtt{m})$$

Source Code Model

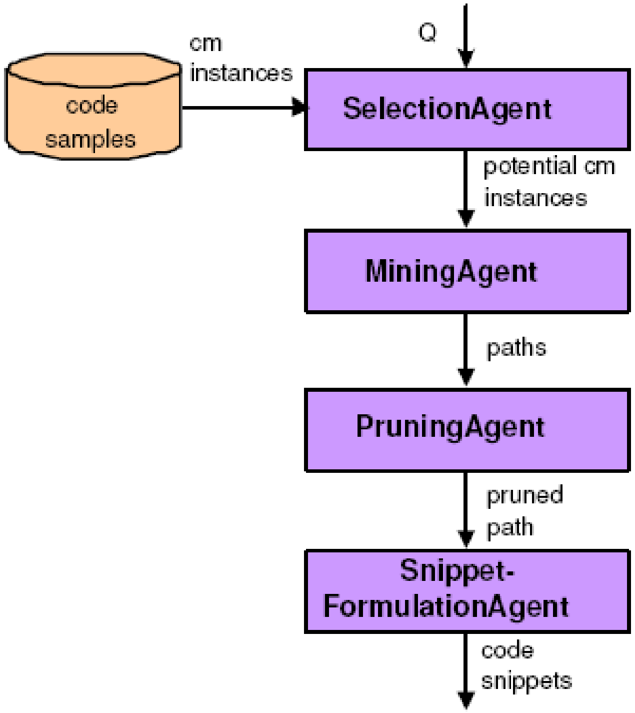
- Only analyse source code text is not sufficient.
- Example:
  - getViewPart().getWorkbenchWindow().getSelectionService().addSelectionListener(this)
  - In the code, an object of type ISelectionService is instantiated, but the type ISelectionService is not denoted.

Source Code Model

- Node types. (All instantiate objects are recorded)
  - type node, object node, method node
- Edge types
  - Class structure: inheritance, implement, and composite edge.
  - Class behaviour: method, assignment, and parameter edge.

Snippet Mining

- Mining all code snippets that satisfy a given user query Q.



Snippet Mining

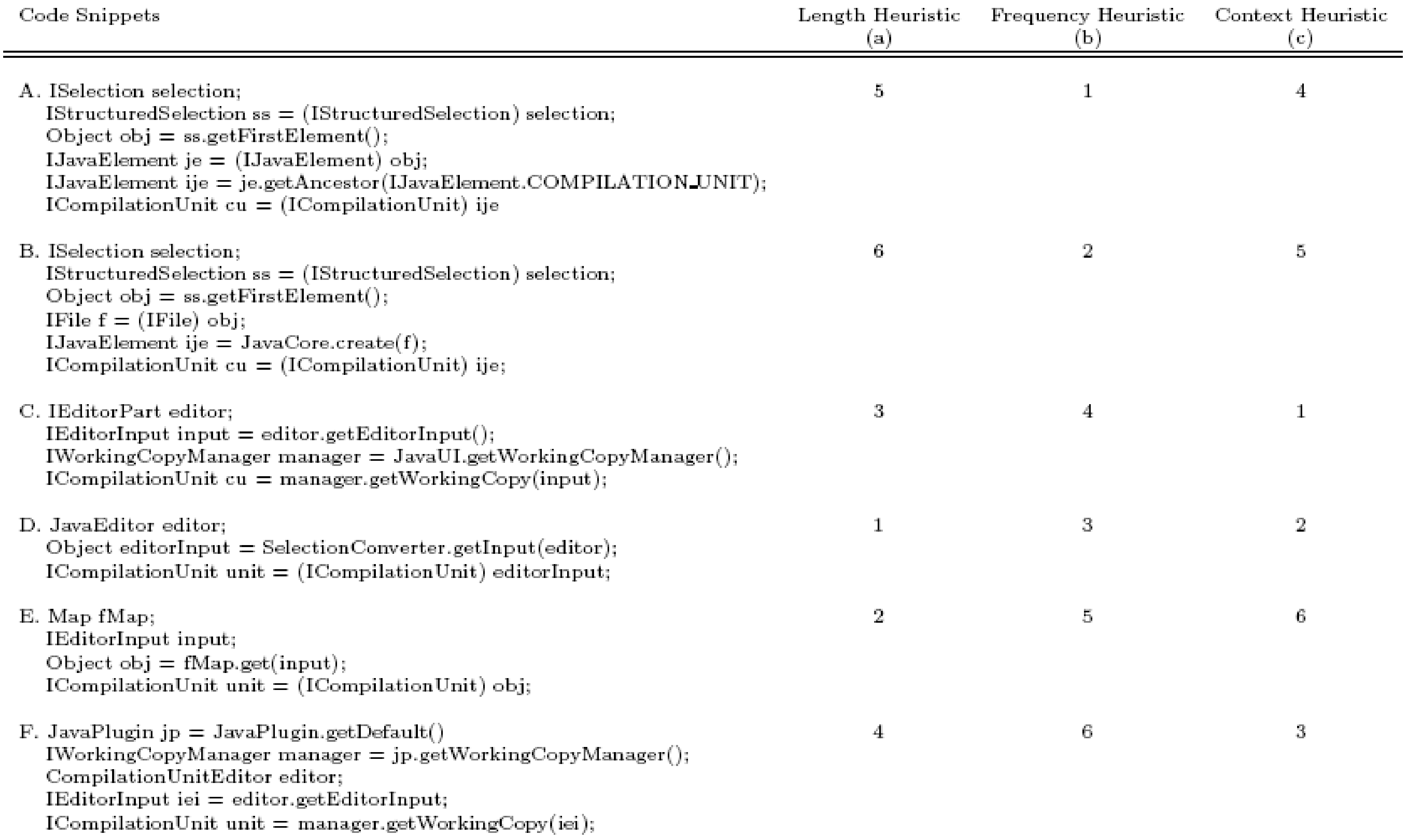
- Selection Agent
  - – Select code models based on input query
- Mining Agent
  - – BFSMINE algorithm
- Pruning Agent
  - – Removes duplicate paths, no-co paths (e.g. IType a = null;), non-compatible and non-executable paths

Ranking Snippets

- Ranking by length
  - Line of code
- Ranking by frequency
  - Number of times a code snippet appears
- Ranking by context
  - Context match measure, Mp match measure between parents, Mct match measure between the lexically visible types

$$\mathcal{M}_{CT}(Q, s) = \frac{\mathcal{M}_{P}(Q, s) + \mathcal{M}_{VT}(Q, s)}{2}$$

Snippets Example



Code Snippets	Length Heuristic (a)	Frequency Heuristic (b)	Context Heuristic (c)

A. ISelection selection;
IStructuredSelection ss = (IStructuredSelection) selection;
Object obj = ss.getFirstElement();
IJavaElement je = (IJavaElement) obj;
IJavaElement ije = je.getAncestor(IJavaElement.COMPILATION_UNIT);
ICompilationUnit cu = (ICompilationUnit) ije
	5	1	4

B. ISelection selection;
IStructuredSelection ss = (IStructuredSelection) selection;
Object obj = ss.getFirstElement();
IFile f = (IFile) obj;
IJavaElement ije = JavaCore.create(f);
ICompilationUnit cu = (ICompilationUnit) ije;
	6	2	5

C. IEditorPart editor;
IEditorInput input = editor.getEditorInput();
IWorkingCopyManager manager = JavaUI.getWorkingCopyManager();
ICompilationUnit cu = manager.getWorkingCopy(input);
	3	4	1

D. JavaEditor editor;
Object editorInput = SelectionConverter.getInput(editor);
ICompilationUnit unit = (ICompilationUnit) editorInput;
	1	3	2

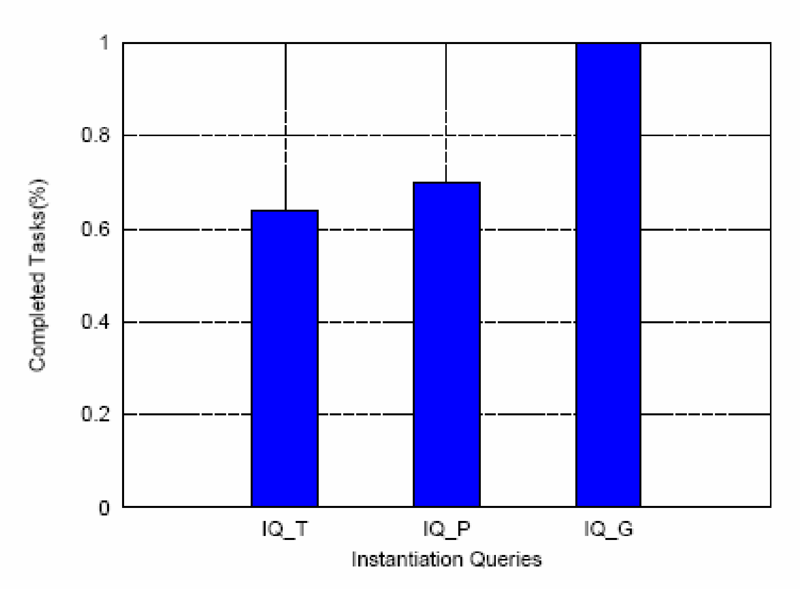
E. Map fMap;
IEditorInput input;
Object obj = fMap.get(input);
ICompilationUnit unit = (ICompilationUnit) obj;
	2	5	6

F. JavaPlugin jp = JavaPlugin.getDefault()
IWorkingCopyManager manager = jp.getWorkingCopyManager();
CompilationUnitEditor editor;
IEditorInput iei = editor.getEditorInput;
ICompilationUnit unit = manager.getWorkingCopy(iei);
	4	6	3

Experiment

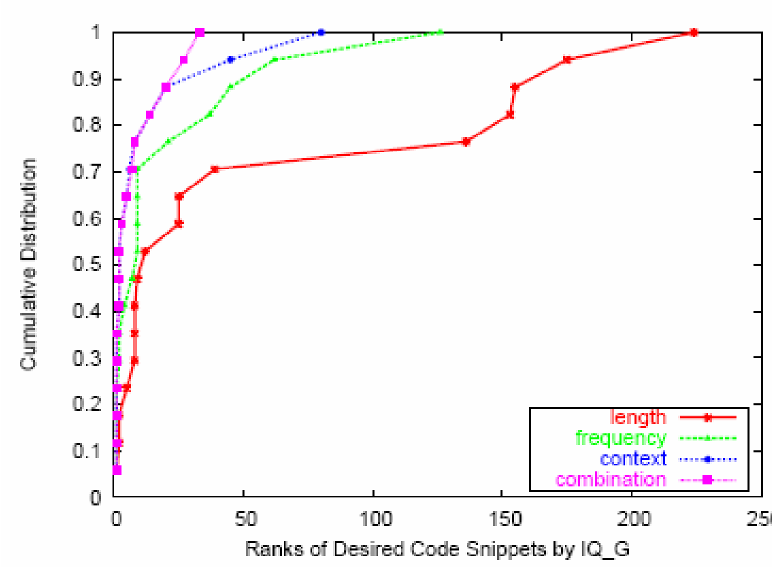
- Hypothesis 1

  Generalized queries provides better coverage of tasks than specialized query. (type and parent based)



Experiment

- Hypothesis 2:
  - Context-sensitive ranking heuristic provides better ranks for best-fit code snippets than context independent heuristics. Similarly, context-independent heuristic degrade sharply with the increase in repository size



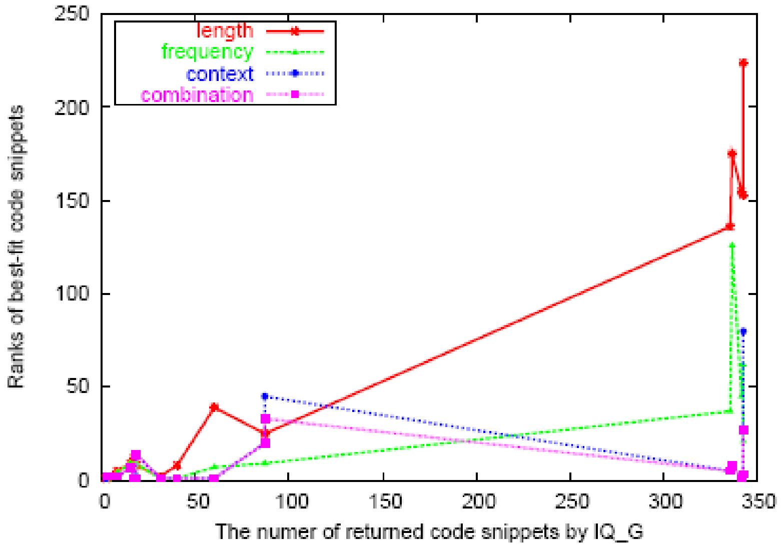
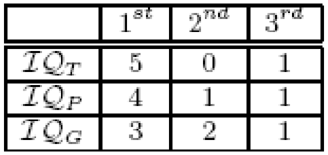


Figure 23: Variation in the Average Rank for the Best-Fit Code Snippet with Increasing Code Snippet Results. Results reported for $\mathcal{IQ}_G$.

Experiment

- Hypothesis 3:
  - Specialized queries combined with context-sensitive ranking heuristics provide better rank ordering for best-fit code snippets than generalized queries using context-sensitive ranking heuristics



	1^{st}	2^{nd}	3^{rd}
$\mathcal{I}Q_T$	5	0	1
$\mathcal{I}Q_P$	4	1	1
$\mathcal{I}Q_G$	3	2	1

Table 2: Distribution of the Best-Fit Ranks for Different Query Types.

Experiment

- Hypothesis 4: The XSnippet system provides significant assistance to developers, enabling them to efficiently complete a large variety of programming tasks.
- Hypothesis 5: The context-dependent approach of the XSnippet system allows developers to complete more tasks than other previously proposed approaches.

Feedback

- Positive
  - Mining both explicit and implicit object instantiations
  - Provides useful snippets
  - Improves developers performance
- Negative
  - BFSMINE section is not easy to understand
  - ?