Clustering and Structuring Code Search Results

Lisa Hua
11/10/2015

Problem

- Code Search Engine returns hundreds of examples based on text matching.
- Within code example results, queried features are interleaving with auxiliary features.
- A single code example hardly contains all expected functionalities.

Motivating Example

- Query: undo redo action in TextEditor

User context:

```
import com.sun.java.swing.*;

public class MyTextEditor {
     public void init() {
    JFrame frame = new JFrame("Undo Sample");
    JTextArea textArea = new JTextArea();
    JButton undoBtn_;

    //add undo and redo action to text editor
}}
```

Motivating Example

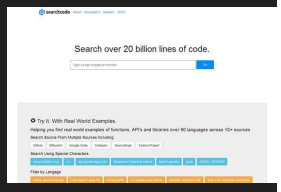
- Query: undo redo action in TextEditor

User context:

```
import com.sun.java.swing.*;

public class MyTextEditor {
     public void init() {
    JFrame frame = new JFrame("Undo Sample");
    JTextArea textArea = new JTextArea();
    JButton undoBtn_;

    //add undo and redo action to text editor
}}
```



Motivating Example

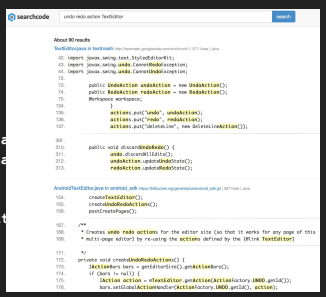
- Query: undo redo action in TextEditor

User context:

```
import com.sun.java.swing.*;

public class MyTextEditor {
     public void init() {
    JFrame frame = new JFrame("Undo Sa
    JTextArea textArea = new JTextArea
    JButton undoBtn_;

    //add undo and redo action to text
}}
```



Motivating Example

- Query: undo redo action in TextEditor

First Example: 960 LOC, 22 inner class, around 60 methods

```
import com.sun.java.swing.*;

public class TextEditor extends JTextPane {

  public UndoAction undoAction = new UndoAction();
  public RedoAction redoAction = new RedoAction();
  public CompoundUndoManager undo;
  HashMap<Object, Action> actions = new
HashMap<Object, Action>();

  public TextEditor(Workspace workspace) {
              this.workspace = workspace;

    undo = new CompoundUndoManager(workspace);
    actions.put("undo", undoAction);
    actions.put("redo", redoAction); }

  public void discardUndoRedo() {
     undo.discardAllEdits();
     undoAction.updateUndoState();
     redoAction.updateRedoState();

}
```

```
//Two inner classes:
public class UndoAction extends
AbstractAction {
  public void actionPerformed(ActionEvent
e) {
     try {
       undo.undo();
       updateUndoState();
       redoAction.updateRedoState();
     } catch (CannotUndoException ex) { }
  }
  public void updateUndoState() {
     setEnabled(undo.canUndo());
  }}
public class RedoAction extends
AbstractAction {
  public void actionPerformed(ActionEvent
e) {
     try {
       undo.redo();
       updateRedoState();
       undoAction.updateUndoState();
     } catch (CannotRedoException ex) {}
  }}
```

Motivating Example

- Query: undo redo action in TextEditor

First Example: 960 LOC, 22 inner class, around 60 methods

```
import com.sun.java.swing.*;

public class TextEditor extends JTextPane {

  public UndoAction undoAction = new UndoAction();
  public RedoAction redoAction = new RedoAction();
  public CompoundUndoManager undo;
  HashMap<Object, Action> actions = new
HashMap<Object, Action>();

  public TextEditor(Workspace workspace) {
              this.workspace = workspace;

     undo = new CompoundUndoManager(workspace);
     actions.put("undo", undoAction);
     actions.put("redo", redoAction); }

  public void discardUndoRedo() {
     undo.discardAllEdits();
     undoAction.updateUndoState();
     redoAction.updateRedoState();
}
```

```
//Two inner classes:
public class UndoAction extends
AbstractAction {
  public void actionPerformed(ActionEvent
e) {
     try {
       undo.undo();
       updateUndoState();
       redoAction.updateRedoState();
     } catch (CannotUndoException ex) {
}}

  public void updateUndoState() {
     setEnabled(undo.canUndo());
  }

public class RedoAction extends
AbstractAction {
  public void actionPerformed(ActionEvent

                                    ;
                                UndoState();
     } catch (CannotRedoException ex) {}}
```

Do I need 'HashMap actions'?
What is 'CompoundUndoManager'?

Motivating Example

- Query: undo redo action in TextEditor

User context:

```
import com.sun.java.swing.*;

public class MyTextEditor {
     public void init() {
     JFrame frame = new JFrame("Undo
Sample");
     JTextArea textArea = new JTextArea();
     JButton undoBtn_;

     //add undo and redo action to text
editor
}}
```

- Do I need 'HashMap actions'?
  - I don't think it is useful in my case
- What is 'CompoundUndoManager'?
  - Query 'CompoundUndoManager' is a subclass of java.swing.UndoManager.

But, …

- How can I use a button to trigger undo/redo action?
- How can I connect UndoManager to my JTextArea?

I need more similar examples...

Motivating Example

- Query: undo redo action in TextEditor

Fourth Example: 223 LOC, 6 inner class, 29 methods

```
import com.sun.java.swing.*;

public class ConsoleTextEditor extends
JScrollPane {
  private UndoAction undoAction = new
UndoAction();
  private RedoAction redoAction = new
RedoAction();
  private TextUndoManager undoManager;
  private TextEditor textEditor = new TextEditor
(...){...}

  public ConsoleTextEditor () {
     this.undoManager = new TextUndoManager();
     Document doc = textEditor.getDocument();
     doc.addUndoableEditListener(undoManager);
     undoManager.addPropertyChangeListener
(undoAction);
     undoManager.addPropertyChangeListener
(redoAction);
     doc.addDocumentListener(undoAction);
     doc.addDocumentListener(redoAction);
}
```

```
//Two inner classes:
private class RedoAction extends
UpdateCaretListener
implements PropertyChangeListener {
  public void actionPerformed(ActionEvent ae)
{
     undoManager.redo();
     setEnabled(undoManager.canRedo());
     undoAction.setEnabled(undoManager.
canUndo());
     }
  public void propertyChange
(PropertyChangeEvent pce) {
     setEnabled(undoManager.canRedo());
}}
private class UndoAction extends
UpdateCaretListener
implements PropertyChangeListener {
  public void actionPerformed(ActionEvent ae)
{
     undoManager.undo();
     setEnabled(undoManager.canUndo());
     redoAction.setEnabled(undoManager.
canRedo());
   }
  public void propertyChange
(PropertyChangeEvent pce) {
     setEnabled(undoManager.canUndo()); }}
 private abstract class UpdateCaretListener
```

Motivating Example

- Query: undo redo action in TextEditor

Fourth Example: 223 LOC, 6 inner class, 29 methods

```
import com.sun.java.swing.*;

public class ConsoleTextEditor extends
JScrollPane {
  private UndoAction undoAction = new
UndoAction();
  private RedoAction redoAction = new
RedoAction();
  private TextUndoManager undoManager;
  private TextEditor textEditor = new TextEditor
(...){...}

  public ConsoleTextEditor () {
     this.undoManager = new TextUndoManager();
     Document doc = textEditor.getDocument();
     doc.addUndoableEditListener(undoManager);
     undoManager.addPropertyChangeListener
(undoAction);
     undoManager.addPropertyChangeListener
(redoAction);
     doc.addDocumentListener(undoAction);
     doc.addDocumentListener(redoAction);
}
```

```
//Two inner classes:
private class RedoAction extends
UpdateCaretListener
implements PropertyChangeListener {
  public void actionPerformed(ActionEvent ae)
{
     undoManager.redo();
     setEnabled(undoManager.canRedo());
     undoAction.setEnabled(undoManager.
canUndo());
     }
  public void propertyChange
(PropertyChangeEvent pce) {
     setEnabled(undoManager.canRedo());
}}
private class UndoAction extends
UpdateCaretListener
implements PropertyChangeListener {
  public void actionPerformed(ActionEvent ae)
{
     undoManager.undo();
     setEnabled(undoManager.canUndo());
     redoAction.setEnabled(undoManager.
canRedo());
   }
  public void propertyChange
(PropertyChangeEvent pce) {
     setEnabled(undoManager.canUndo());
```

Motivating Example - Cont.

There are multiple ways to finish this reuse task.

Example 2: android --- AndroidTextEditor	Example 5: Eclipse --- PaparusCDTditor
`IActionBars bars = getEditorSite().getActionBars(); if (bars != null) { IAction action = textEditor.getAction (ActionFactory.UNDO.getId()); bars.setGlobalActionHandler(ActionFactory.UNDO.getId(),action); action = mTextEditor.getAction(ActionFactory.REDO.getId()); bars.setGlobalActionHandler (ActionFactory.REDO.getId(), action); bars.updateActionBars(); }`	`IActionBars actionBars = getActionBars(); if (actionBars != null) { IAction textUndo = actionBars.getGlobalActionHandler (ITextEditorActionConstants.UNDO); IAction textRedo = actionBars.getGlobalActionHandler (ITextEditorActionConstants.REDO); actionBars.setGlobalActionHandler (ITextEditorActionConstants.UNDO, textUndo); actionBars.setGlobalActionHandler (ITextEditorActionConstants.REDO, textRedo); actionBars.updateActionBars();}`
Example 8: sdk --- LayoutCanvas	Example 6: Eclipse-Plugin --- AspectEditorContributor

Hypothesis:

- Code Search Engine returns hundreds of examples.

→ Clustering code examples may help user identify the examples they prefer.

- Queried features are interleaving with auxiliary features.

→ Multiple similar examples may help discover common features.

- A single code example hardly contains all expected functionalities.

→ A cluster of similar examples may help identify useful auxiliary features.

Hypothesis:

- Code Search Engine returns hundreds of examples.

→ Clustering code examples may help user identify the examples they prefer.

Clustering similar examples and finding the commonality may help remove irrelevant parts.

- A single code example hardly contains all expected functionalities.

→ A cluster of similar examples may help identify useful auxiliary features.

Approach

- Extract query-related structural facts from source code
- Match facts for each example pair
- Cluster examples hieratically

Approach - Step 1: Extract query-related structural fact

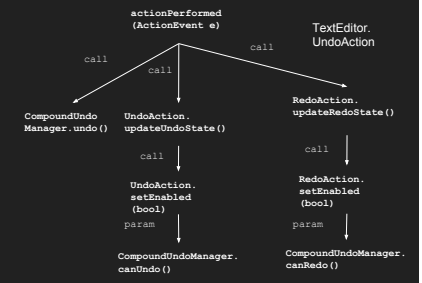
1. Locate query-related feature based on IR:
   - LSI: Latent Semantic Index. Extract each method and query as a term vector, use matrix to compare the similarity.
2. Extract structural facts as ontology instances --- (subject, predicate, object)

	Type	Description
Class level	(c1, subType, c2) (c1, innerclass, c2) (c, field, v) (c, method, m)	class c1 extends c2 class c1 has an inner class c2 class c has field v class c contains method m
Method Level	(m, parameter, v) (m, return, t) (e1, accesses, e2) (m1, call, m2) (m, variable, v)	method m has parameter v method m returns an object of type t e1 = e2, e1 and e2 are expressions method m1 calls method m2 method m has variable m2
	(o, name, s)	The name of object o is s

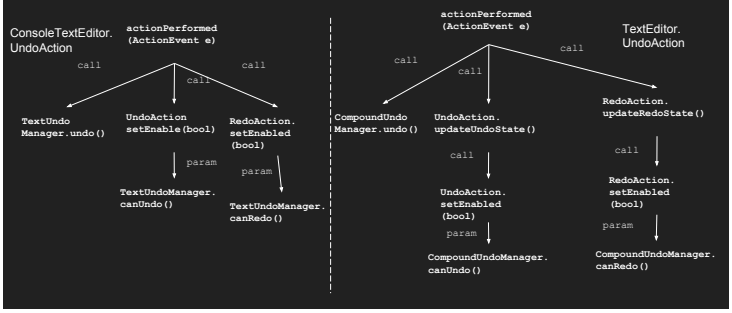
Approach - Step 1: Extract query-related structural fact

1. Locate query-related feature based on IR.
2. Extract structural facts in an ontology instance --- (subject, predicate, object)
3. Manipulate the relationship between different facts using Resource Description Framework (RDF). Each (subject, predicate, object) triple is an edge from subject to object labeled with predicate.

```
public class UndoAction extends
AbstractAction {
  public void actionPerformed
(ActionEvent e) {
     undo.undo();
     updateUndoState();
     redoAction.updateRedoState();
  }
  public void updateUndoState() {
     setEnabled(undo.canUndo());
  }}
```



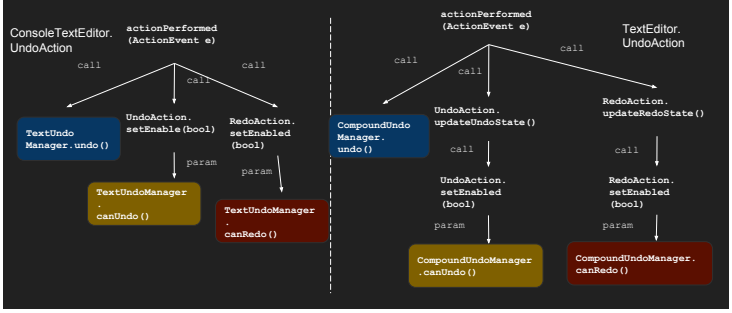
Approach - Step 2: Match facts for each example pair



Approach - Step 2: Match facts for each example pair

Cupid: Generic Schema Matching with Cupid. Jayant Madhavan et al. VLDB'01.

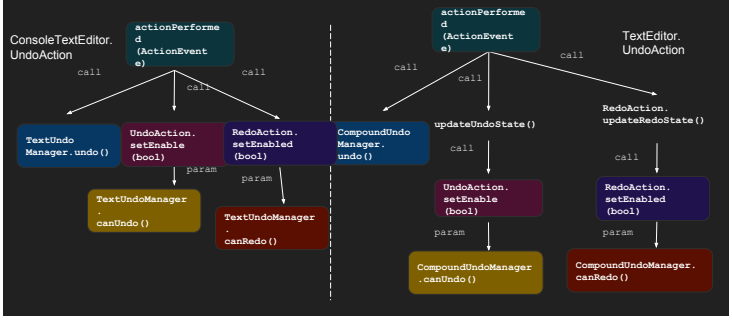
Idea: The facts without outer edge represent the atomic functionality. Two RDF elements are structurally similar if their leaf sets are highly similar.



Approach - Step 2: Match facts for each example pair

Cupid: Generic Schema Matching with Cupid. Jayant Madhavan et al. VLDB'01.

Idea: The facts without outer edge represent the atomic functionality. Two RDF elements are structurally similar if their leaf sets are highly similar.



Approach - Step 2: Match facts for each example pair

Cupid: Generic Schema Matching with Cupid. Jayant Madhavan et al. VLDB'01.

$w_{sim} = w_{struct} \times s_{sim} + (1-w_{struct}) \times l_{sim}$ w_{sim} : weighted similarity, w_{struct} : constant from 0 to 1, l_{sim} : Linguistic similarity, s_{sim} : Structural similarity,	Step 1: calculate l_{sim} for leaf nodes Step 2: calculate s_{sim} and w_{sim} for non-leaf nodes TreeMatch(SourceTree S, TargetTree T) S' = post-order(S), T' = post-order(T) for each s in S' for each t in T' $w_{sim}(s,t) = w_{struct} \cdot s_{sim}(s,t) + (1-w_{struct}) \cdot l_{sim}(s,t)$ if $w_{sim}(s,t) > th_{high}$ increase-struct-similarity (leaves(s),leaves(t),c_{inc}) if $w_{sim}(s,t) < th_{low}$ decrease-struct-similarity(leaves(s),leaves(t), c_{dec})	$s_{sim}(s,t) =$ $\{x \mid x \in leaves(s) \wedge \exists y \in leaves(t), stronglink(x, y)\}$ $\cup \{x \mid x \in leaves(t) \wedge \exists y \in leaves(s), stronglink(y, x)\}$ / $\mid leaves(s) \cup leaves(t) \mid$ th_{high}, th_{low}, c_{inc} ,c_{dec}: if two leaves' weighted similarity exceeds the threshold *thhigh*, we increase the structural similarity (s_{sim}) of each pair of leaves in the two subtrees by the factor *cinc* *Two elements are similar (stronglink) iff wsim(s, t) ≥ thaccept* mapping may be 1:n.

Approach - Step 3: Cluster examples hierarchically

Approach - Step 3: Cluster examples hierarchically

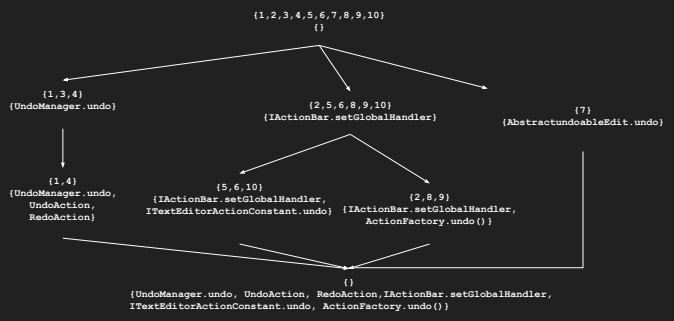
Example:

1	textmash --- TextEditor	CompoundUndoManager.undo, CompoundUndoManager.redo, UndoAction, RedoAction
2	android-AndroidTextEditor	IActionBars.setGlobalActionHandler(), ActionFactory.UNDO, ActionFactory.REDO
3	ide-DocumentUndoImp	FileDocumentUndoManager.undo, FileDocumentUndoManager.redo
4	groovy-ConsoleTextEditor	TextUndoManager.undo, TextUndoManager.redo, UndoAction, RedoAction
5	Eclipse-PapyrusCDTEditor	IActionBars.setGlobalActionHandler(), ITextEditorActionConstants.UNDO
6	Eclipse-AspectEditorContributor	IActionBars.setGlobalActionHandler(), ITextEditorActionConstants.UNDO
7	antlrworks-ATETextPane	AbstractUndoableEdit.undo(), AbstractUndoableEdit.redo(),
8	sdk-LayoutCanvas	IActionBars.setGlobalActionHandler(), ActionFactory.UNDO, ActionFactory.REDO
9	bukminster-IUEditorContributor	IActionBars.setGlobalActionHandler(), ActionFactory.UNDO, ActionFactory.REDO
10	Eclipse-WSDLContributor	IActionBars.setGlobalActionHandler(), ITextEditorActionConstants.UNDO

Approach - Step 3: Cluster examples hierarchically

Formal Concept Analysis:

Formal context is a triple K = (G, M, I) where G is a set of objects, M is a set of attributes, the binary relation I ⊆ G × M.
In our case, objects are code examples, attributes are structural facts.



Approach - Step 3: Cluster examples hierarchically

Formal Concept Analysis:

Formal context is a triple K = (G, M, I) where G is a set of objects, M is a set of attributes, the binary relation I ⊆ G × M.
In our case, objects are code examples, attributes are structural facts.

The concepts (G_i, M_i) defined above can be partially ordered by inclusion: if (G_i, M_i) and (G_j, M_j) are concepts, we define a partial order ≤ by saying that $(G_i, M_i) \leq (G_j, M_j)$ whenever $G_i \subseteq G_j$. Equivalently, $(G_i, M_i) \leq (G_j, M_j)$ whenever $M_j \subseteq M_i$.

Preliminary Results and Evaluation Dataset

	Query	# Examples	Avg LOC	Avg Extracted LOC	First-Level Clusters
1	undo redo TextEditor	68	477	107	6
2	copy paste data from clipboard	24	886	178	5
3	open url html browser	426	780	68	7
4	track mouse hover	33	770	177	4
5	open file external editor	341	927	192	7
6	batch workspace changes single operation	11	928	192	3
7	remove problem marker from resource	211	664	206	8
8	highlight text range editor	111	293	219	6
9	update status line	445	921	102	7
10	prompt user select dictionary	114	893	129	5

Related Works

- Code search
- Identify structural correspondence
- Feature location

Conclusion

- It is hard to identify a perfect code example containing all desired features from hundreds of code search results.
- We provide an approach to cluster and structure code search results hierarchically, to extract common features and specific features.
- Our approach may ease code reuse.