Information Retrieval Applications in Software Engineering

Sonia Haiduc

Assistant Professor
Department of Computer Science
Florida State University

Short Bio

What is Software Engineering?

How about Software Engineering Research?

Information Retrieval Applications in Software Engineering

Sonia Haiduc

Assistant Professor
Department of Computer Science
Florida State University

What is Information Retrieval?

SE Tasks Supported by Information Retrieval

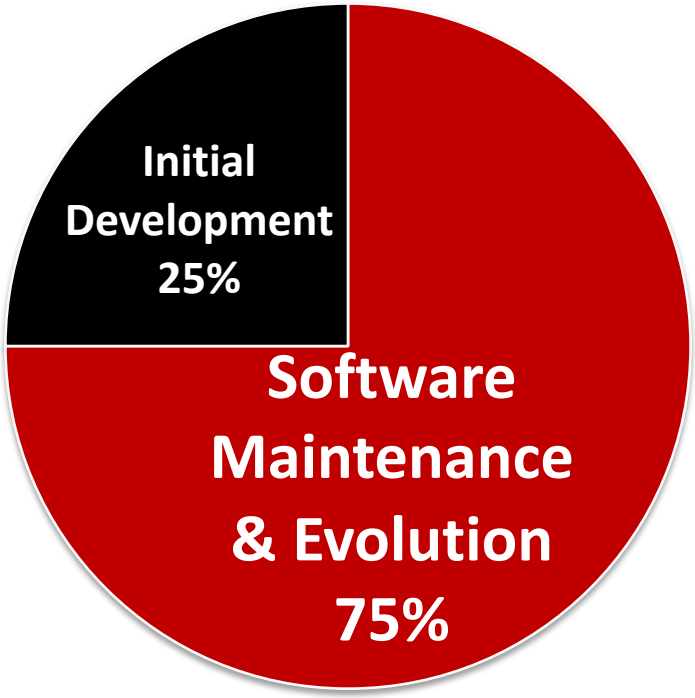
- Concept/Feature Location
- Impact Analysis
- Traceability Link Recovery
- Code Reuse
- Bug Triage
- Program Comprehension
- Architecture/design recovery
- Quality Assessment
- Software Evolution Analysis
- Automatic Documentation
- Requirements Analysis
- Defect Prediction and Debugging
- Refactoring
- Software Categorization
- Licensing Analysis
- Clone Detection
- Effort Estimation
- Domain Analysis
- Web Services Discovery

SE Tasks Supported by Information Retrieval

- **Concept/Feature Location**
- Impact Analysis
- Traceability Link Recovery
- Code Reuse
- Bug Triage
- Program Comprehension
- Architecture/design recovery
- Quality Assessment
- Software Evolution Analysis
- Automatic Documentation
- Requirements Analysis
- Defect Prediction and Debugging
- Refactoring
- Software Categorization
- Licensing Analysis
- Clone Detection
- Effort Estimation
- Domain Analysis
- Web Services Discovery

Software Changes

- Adding new features
- Modifying existing features
- Fixing bugs
- Improving performance
- Adapting to changes in hardware
- Refactoring
- Etc.



Software Costs

Software Change is Difficult

(because software is hard to understand)

- Millions of lines of code
  - S-class Mercedes-Benz : 20 million
  - OpenOffice: 30 million
  - Windows XP: 45 million
- Developed by large, distributed, and diverse teams
- Developers have to change software with:
  - Limited domain knowledge
  - Absence of the original developer
  - Bad, missing, or out of date documentation

Concept Location

- Finding the implementation of a concept in the code, i.e., a place in the source code where to start a change

- Sources of information:
  - *Structure* - the structural aspects of the source code (e.g., control and data flow, class diagrams)
  - *Dynamic* – behavioral aspects of the program (e.g., execution traces)
  - *Text* - captures the problem domain and developer intentions (e.g., identifiers, comments) -> **Text Retrieval**

Text Retrieval for Concept Location

Query

INPUT

TR Engine

#	Method	Class	Score
1	getFace	org.eclipse.ui.JFace	0.99
2	nextEntry	org.eclipse.jdt.IndexBlock	0.96
3	getSeparator	org.eclipse.jdt.core.Util	0.95
4	validate	org.eclipse.jface.IDialog	0.87
5	setTextDlg	org.eclipse.ui.Text	0.86

Relevant Code Elements

Source Code Text

Problems

Query

- Developers have a hard time formulating good queries in unfamiliar software systems

Source Code Text

- The results of TR depend on the quality of identifiers found in the source code

#	Method	Class	Score
1	getFace	org.eclipse.ui.JFace	0.99
2	nextEntry	org.eclipse.jdt.IndexBlock	0.96
3	getSeparator	org.eclipse.jdt.core.Util	0.95
4	validate	org.eclipse.jface.IDialog	0.87
5	setTextDlg	org.eclipse.ui.Text	0.86

Results Presentation

- The presentation of the results does not offer enough information to understand if the results are relevant

Query

Problem #1

Problem

- Developers have a hard time formulating good queries in unfamiliar software systems

Research Questions

- How can query formulation be made easy for developers?
- How can bad queries be improved?

Solution

- Automatic query reformulation

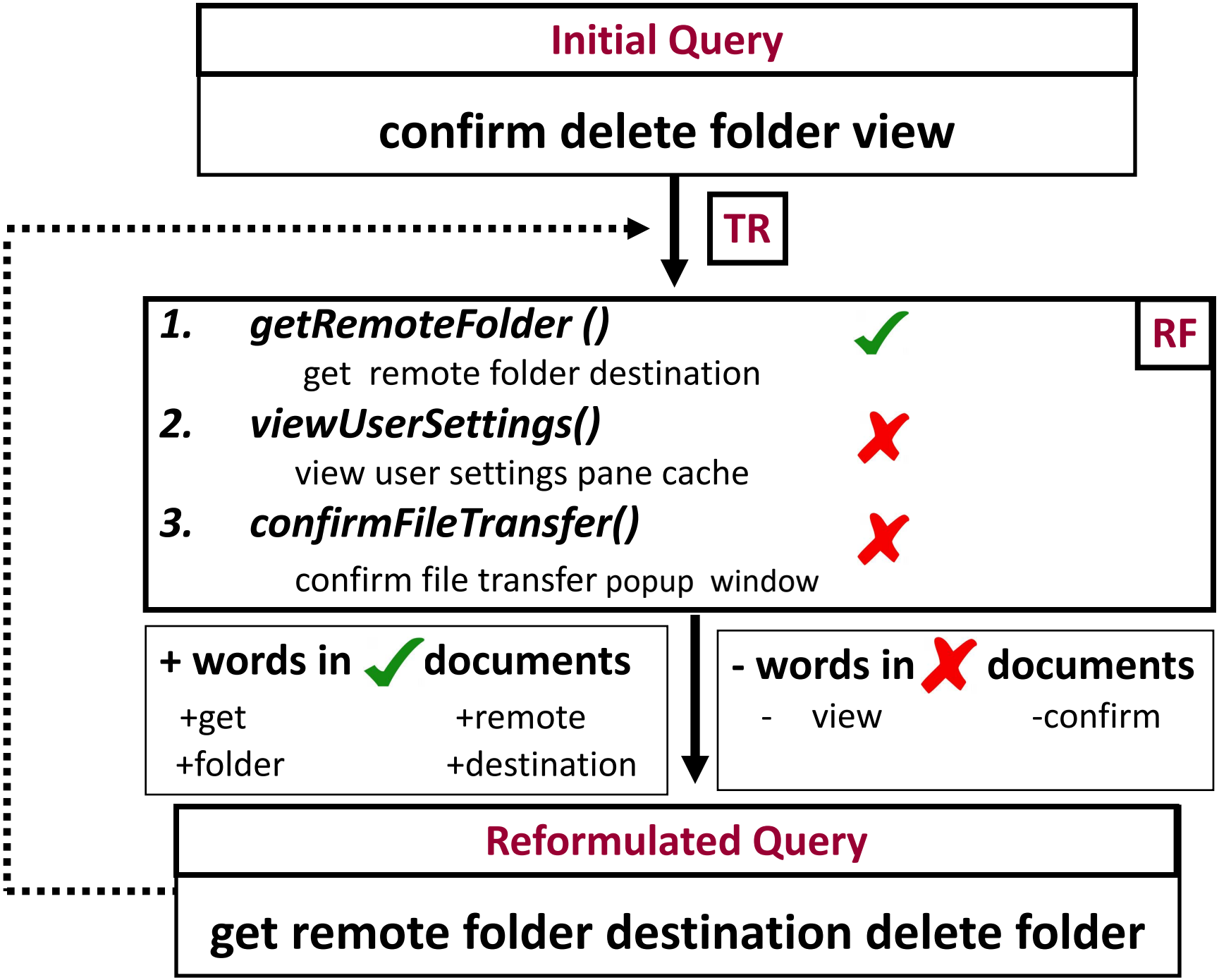
Approaches

- **Semi-automatic**: Relevance feedback
  - People can not always express well what they are looking for, but can recognize it when they see it
  - Developer provides feedback about relevance of search results and query is automatically reformulated

- **Fully automatic**: Learning the best reformulation for each query
  - Developer needs not be involved
  - Use machine learning techniques to learn the best reformulation for queries based on their lexical properties

FileZilla Bug Report #3272

No confirm for delete in folder view
Reported by: trellmor **Priority:** normal **Component:** FileZilla client
Description If you try to delete a folder by “right click -> delete” in the remote folder window, it won’t ask for confirmation.


Initial Query
confirm delete folder view
TR
1. getRemoteFolder ()
get remote folder destination
2. viewUserSettings()
view user settings pane cache
3. confirmFileTransfer()
confirm file transfer popup window
RF
+ words in documents
+get
+folder
+remote
+destination
- words in documents
- view
-confirm
Reformulated Query
get remote folder destination delete folder

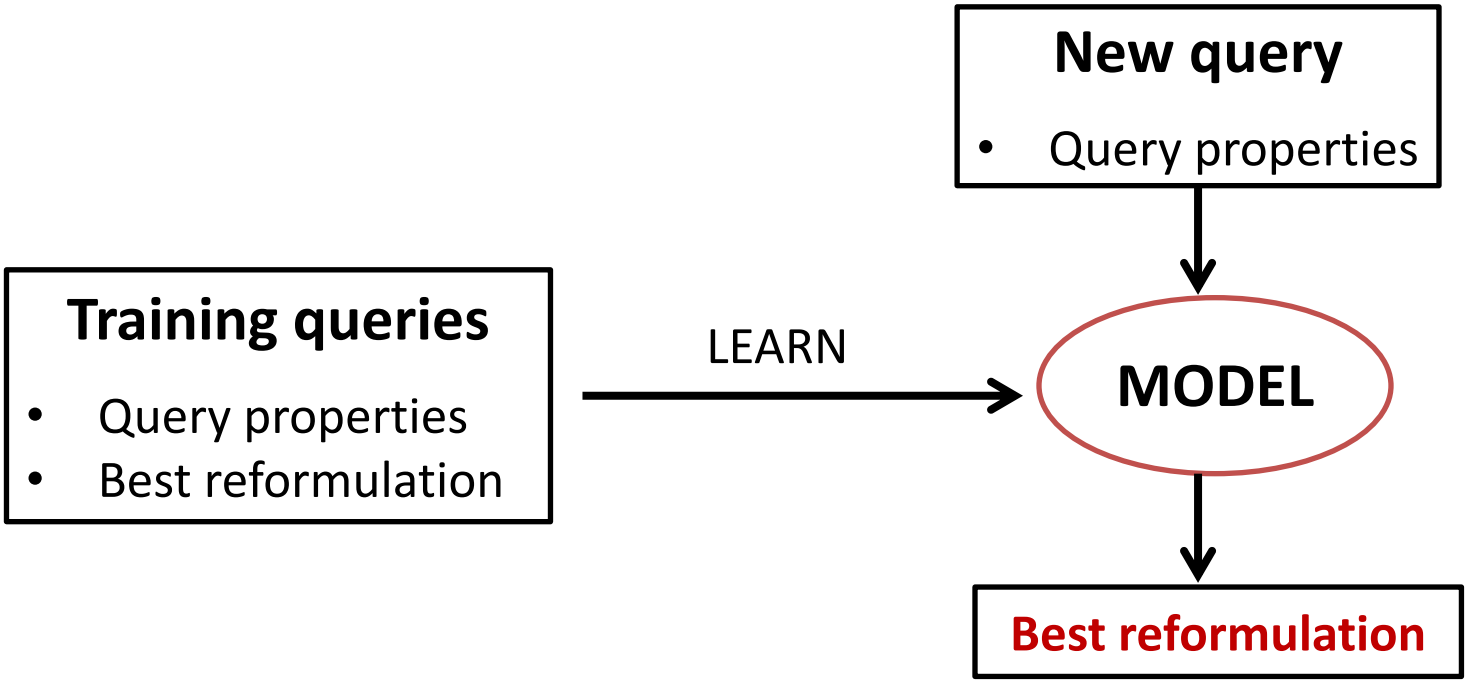
Evaluation

- Empirical evaluation - locating bugs in code based on text found in bug reports
- Patches in bug reports used for identifying buggy methods
- 3 large software systems, 18 queries
  - Eclipse – IDE for Java (2500 KLOC)
  - jEdit – programming editor (300 KLOC)
  - Adempiere – enterprise resource planning (330 KLOC)
- ***Results:*** 72% of cases queries reformulated using relevance feedback led to better results

Refoqus: Automatically Determining the Best Reformulation

- In relevance feedback, developers need to spend time providing feedback - automated solution desirable
- Queries are different - different types of queries may require different reformulation approaches (query expansion, query contraction, etc.)

Refoqus


New query
• Query properties
Training queries
• Query properties
• Best reformulation
LEARN
MODEL
Best reformulation

Evaluation

- Empirical evaluation evaluation - locating bugs in code based on text found in bug reports

- 6 software systems, 30 queries each
  - Adempiere (330 KLOC)
  - Atunes (80 KLOC)
  - FileZilla (240 KLOC)
  - jEdit (300 KLOC)
  - Mahout (110 KLOC)
  - WinMerge (410 KLOC)

- ***Results:*** Refoqus outperformed any individual reformulation technique; 85% of cases improved results of TR-based concept location

Source Code Text

Problem #2

Problem

- The results of TR depend on the quality of identifiers found in the source code

Research Question

- How can we improve the results of TR-based concept location when bad identifiers are present?

Solution

- Identifying and renaming bad identifiers

Lexicon Bad Smells

- Poorly named identifiers can be misleading and impact the results of TR techniques
- Defined a catalog of bad smells in identifiers
- Proposed a set of renaming operations to fix bad smells
- Empirical evaluation on concept location
- ***Results***: improved TR-based concept location after removing bad smells

#	Method	Class	Score
1	getFace	org.eclipse.ui.JFace	0.99
2	nextEntry	org.eclipse.jdt.IndexBlock	0.96
3	getSeparator	org.eclipse.jdt.core.Util	0.95
4	validate	org.eclipse.jface.IDialog	0.87
5	setTextDlg	org.eclipse.ui.Text	0.86

Results Presentation

Problem #3

Problem

- The presentation of the results does not offer enough information to understand if the results are relevant

Research Question

- How can the results of TR-based concept location be presented in a more informative way?

Solution

- Automatic code summaries

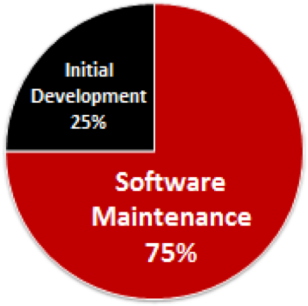
Code Summaries

- Brief but relevant descriptions of source code entities (methods, classes, etc.)
- Text retrieval and text summarization techniques extract most representative information from code
- User evaluation for method and class summaries
- ***Results***: users agreed with the summaries created (score 3.2 out of 4)

- Current work: people summarize code differently - user studies

Software Changes

- Adding new features
- Modifying existing features
- Fixing bugs
- Improving performance
- Adapting to changes in hardware
- Refactoring
- Etc.



Software Costs

Text Retrieval for Concept Location

Query

INPUT

TR Engine

#	Method	Class	Score
1	getFace	org.eclipse.ui.IFace	0.99
2	nextEntry	org.eclipse.jdt.IndentBlock	0.96
3	getSeparator	org.eclipse.jdt.core.Util	0.95
4	validate	org.eclipse.jface.IDialog	0.87
5	setTextDlg	org.eclipse.ui.Text	0.86

Relevant Code Elements

Source Code Text

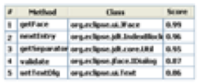
Problems

Query

- Developers have a hard time formulating good queries in unfamiliar software systems

Source Code Text

- The results of TR depend on the quality of identifiers found in the source code



Results Presentation

- The presentation of the results does not offer enough information to understand if the results are relevant

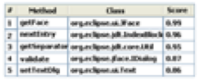
Solutions

Query

- Query reformulation

Source Code Text

- Identifying and renaming bad identifiers



Results Presentation

- Automatic code summaries