

Towards Automatically Generating Summary Comments for Java Methods

Giriprasad Sridhara, Emily Hill, Divya Muppaneni,
Lori Pollock and K. Vijay-Shanker

WILLIAM KILLIAN
CISC 879 - TEXT ANALYSIS IN SOFTWARE ENGINEERING
31 OCTOBER 2013

*“Good **comments** can help **programmers**
quickly understand what a method does...*

*Unfortunately **few projects** adequately
comment code”*

Houston, we have a problem

- Developers don't comment their code
- Comments can help developers better understand code
- We want developers (that's us!) to write understandable code quickly

CHICKEN — EGG PROBLEM

More Problems

- We can't force developers to write comments

Even if we could, what about existing projects?

- Existing techniques can't generate comments well
- We can't get rid of comments by using complete identifiers

*I don't always code but when I do I use identifiers like
updatedBondsRepurchasePrice*

Existing Work

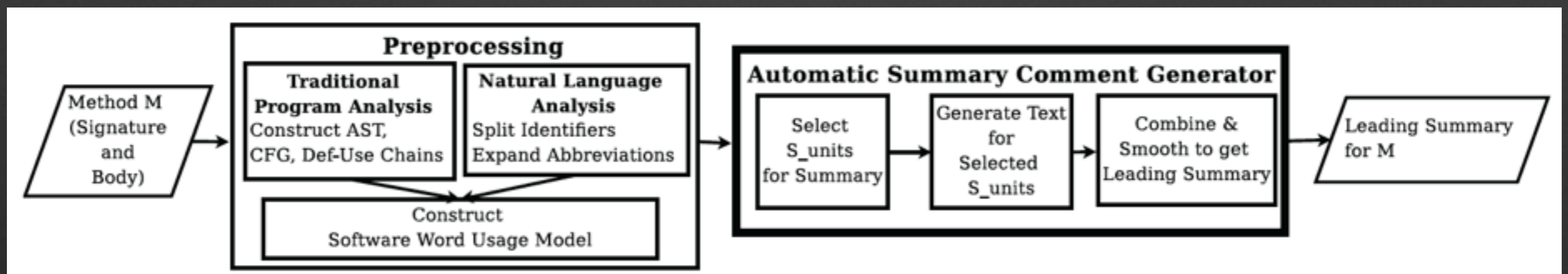
- Prompting developer to write comments
- Documentation-first approach
- Automatically generating comments
 - *Generate comments on exceptions*
- Identifying key statements in code

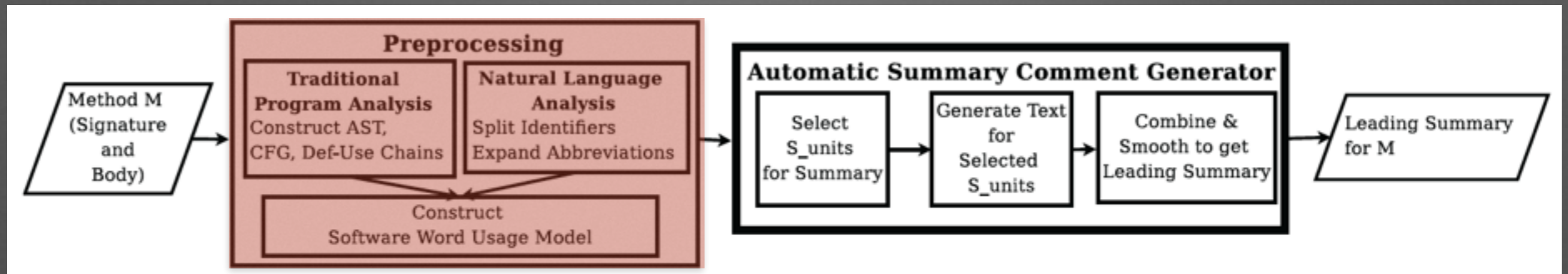
Generating Comments Automatically

- Given a method, M , and its body how can we generate natural language text that describes the overall actions of M
 - **accurately** — is the summary properly defining what the method does
 - **adequately** — does the summary contain all adequate content
 - **concisely** — does the summary minimize redundancy

Solution Breakdown

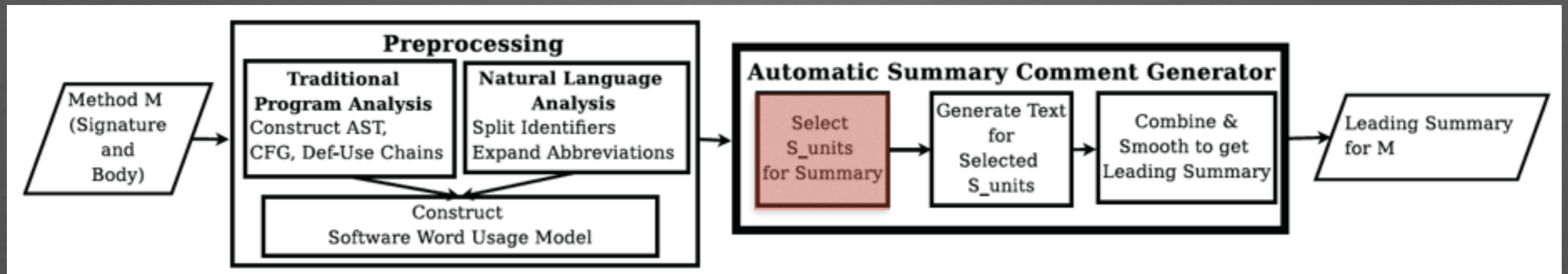
- **Accurately** — Selecting the content (s_units)
- **Adequately** — Lexicalizing and generating natural language text
- **Concisely** — Combining and smoothing generating text





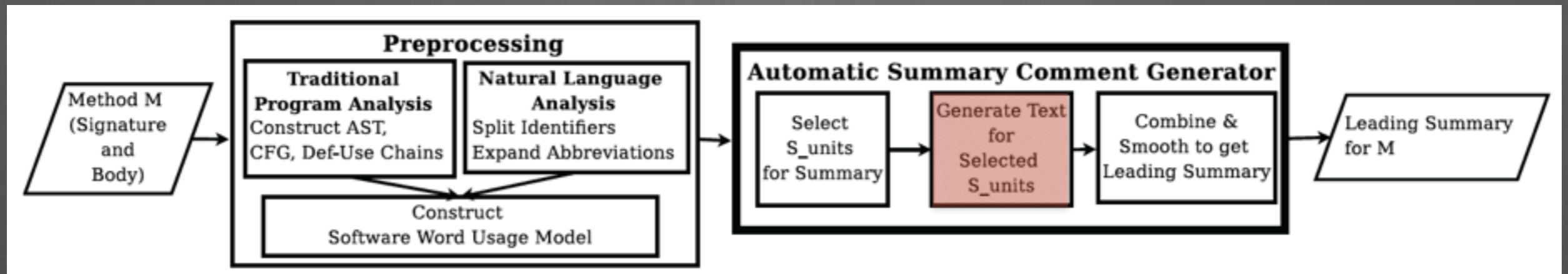
Preprocessing

- Camel-case splitting
- Identify and expand abbreviations
- Construct Software Word Usage Model
 - Identifying action, theme, and secondary arguments
 - methods start with verbs
 - infer context based on location of words with another
 - `saveImage()` `book.toString()`



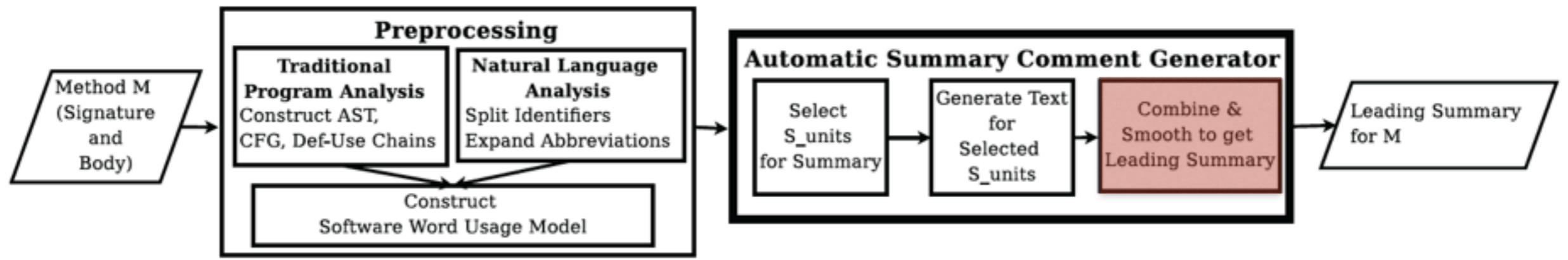
Selecting S_units

- choose the important or central lines of code
- S_unit (in general) is a single java statement
 - Ending — exists at control exit (end)
 - Void-Return — method call performing some action
 - Same-Action — S_unit performs action of method
 - Data-Facilitating — assign data to variables
 - Controlling — conditional branching (if X then Y else Z)



Generate Text

- Construct subphrases in S_units and concatenate
- Lexicalization of Variables
 - identify theme equivalences



Contributions

Extraction Algorithm

- Automatically obtain important code statements for summary

Text Generation Technique

- Java Code ► Natural Language

Human Evaluation

- Measure accuracy, content adequacy, and conciseness

