Leveraging Natural Language Analysis of Software: Achievements, Challenges, and Opportunities

Lori L. Pollock
Computer and Information Sciences
University of Delaware

Software is like a car.

It breaks.

Software is like a car.

We want it to go faster.

Software is like a car.

We want more features.

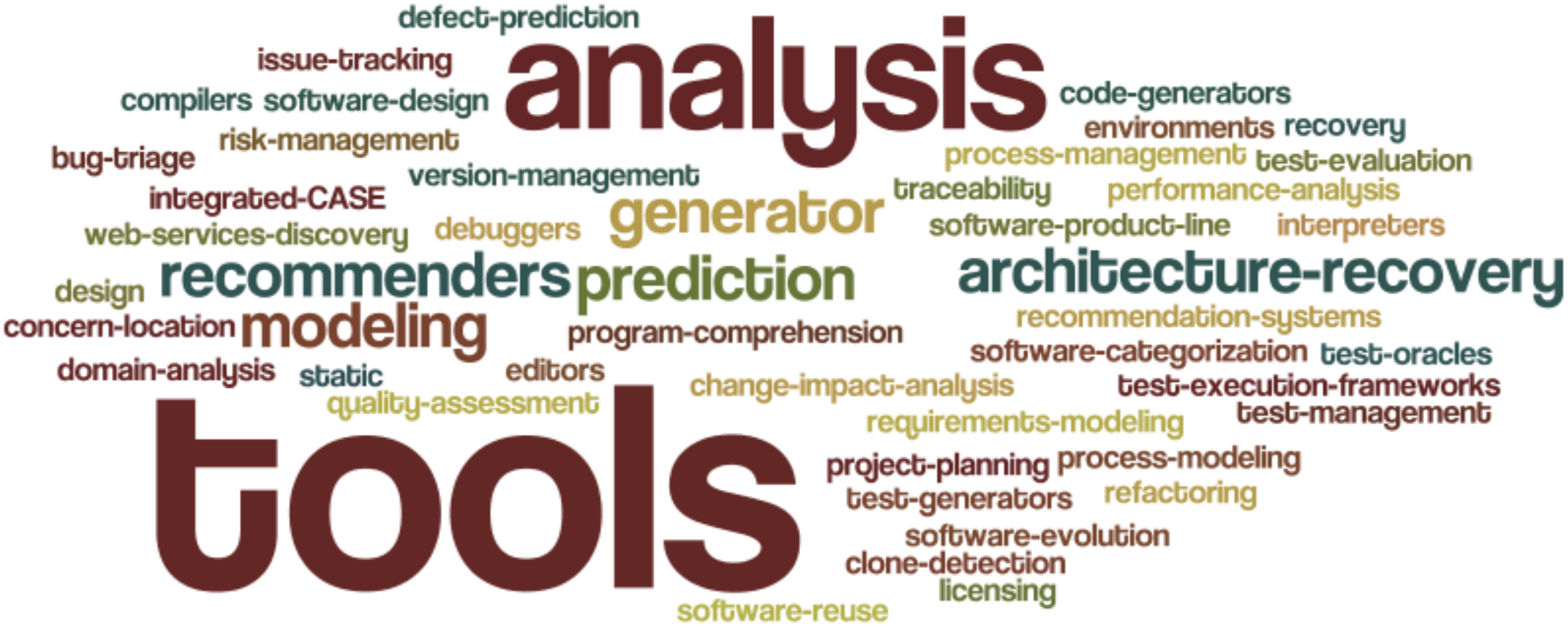
Software is like a car.

It is increasingly complex under the hood.

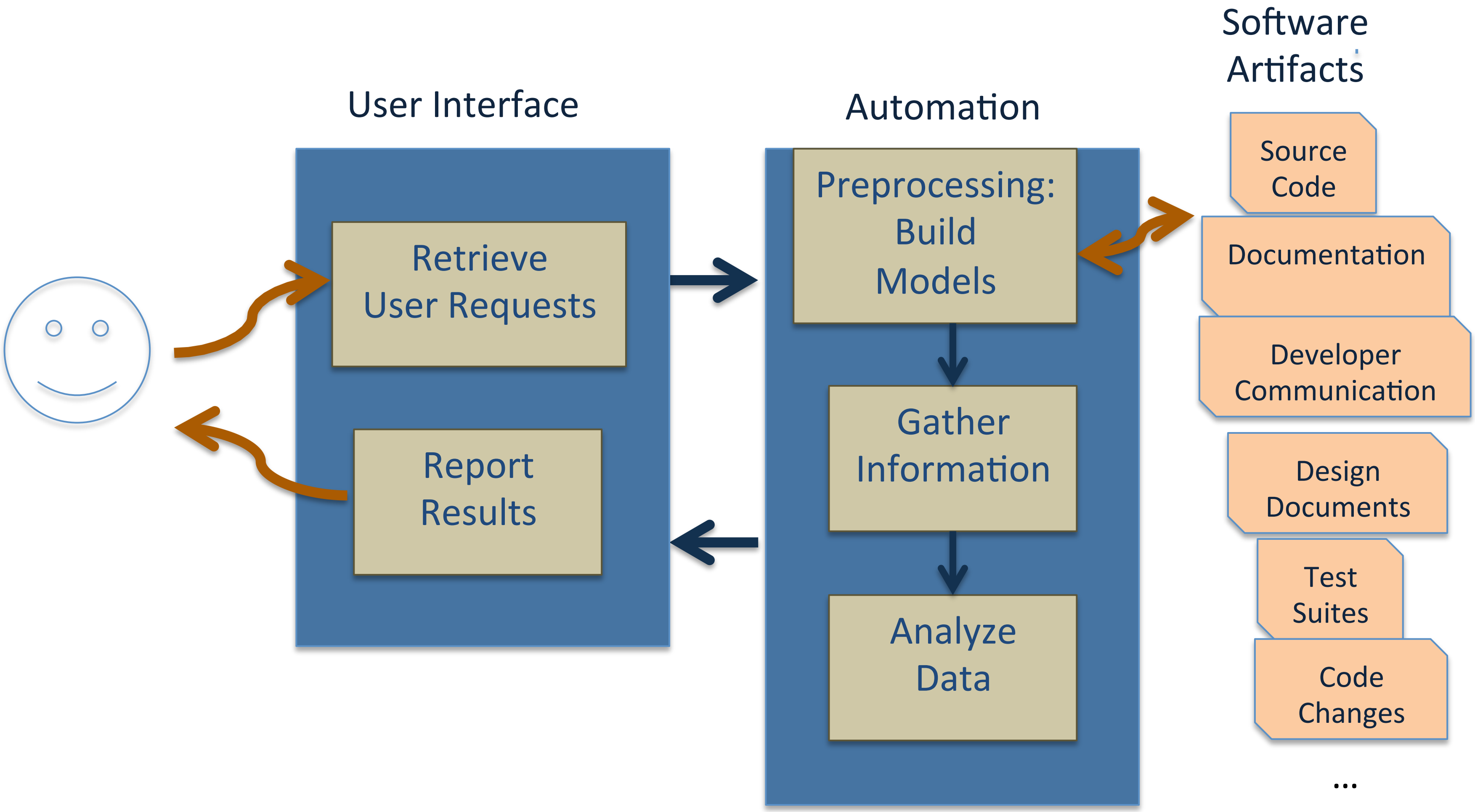
Software is like a car.

It now requires specialized tools to maintain.

SE community to the rescue



Power Tools



User Interface

Retrieve User Requests

Report Results

Automation

Preprocessing: Build Models

Gather Information

Analyze Data

Software Artifacts

Source Code

Documentation

Developer Communication

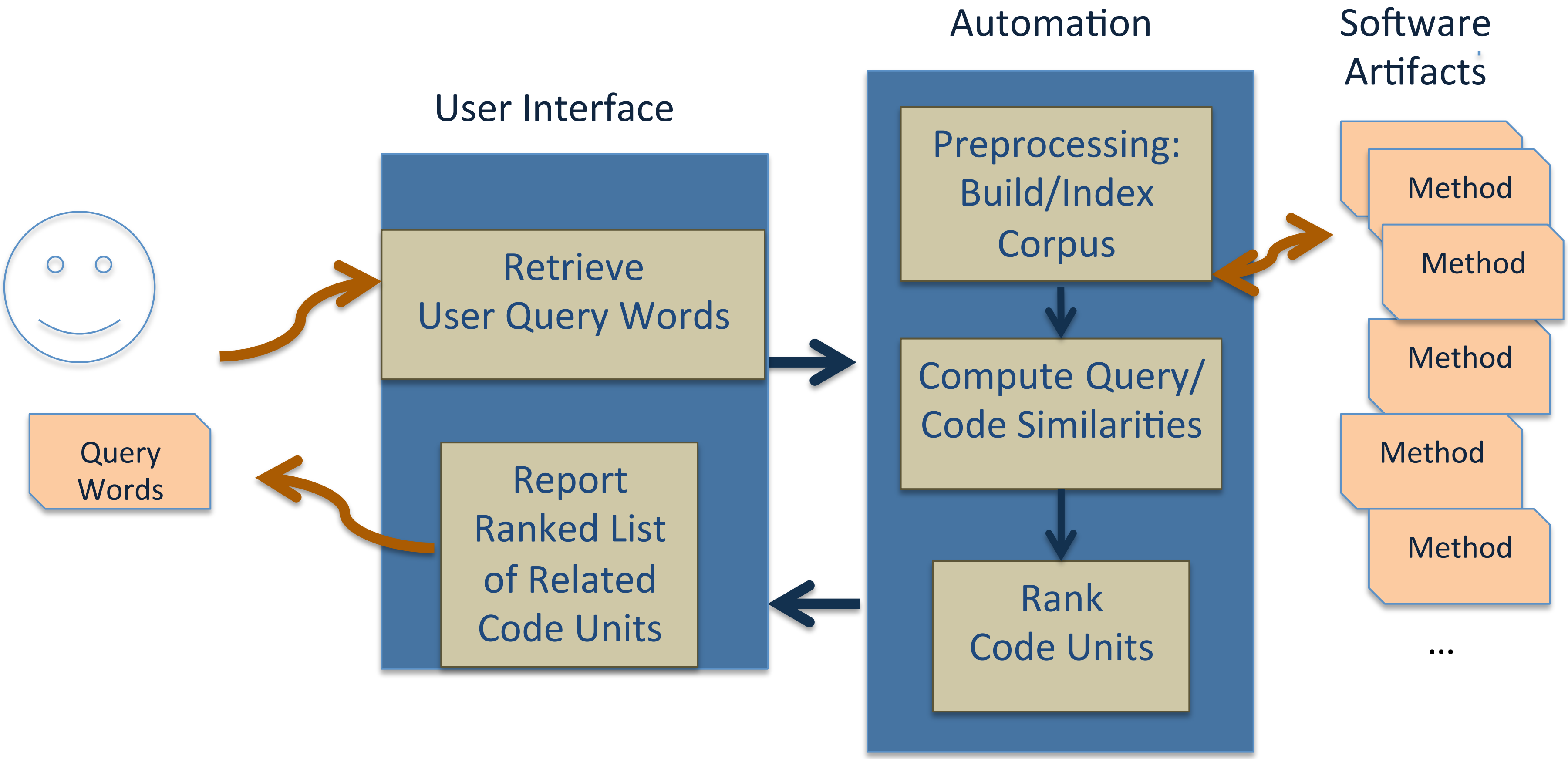
Design Documents

Test Suites

Code Changes

…

Example: Code Search Tool



User Interface

Retrieve User Query Words

Report Ranked List of Related Code Units

Query Words

Automation

Preprocessing: Build/Index Corpus

Compute Query/ Code Similarities

Rank Code Units

Software Artifacts

Method

Method

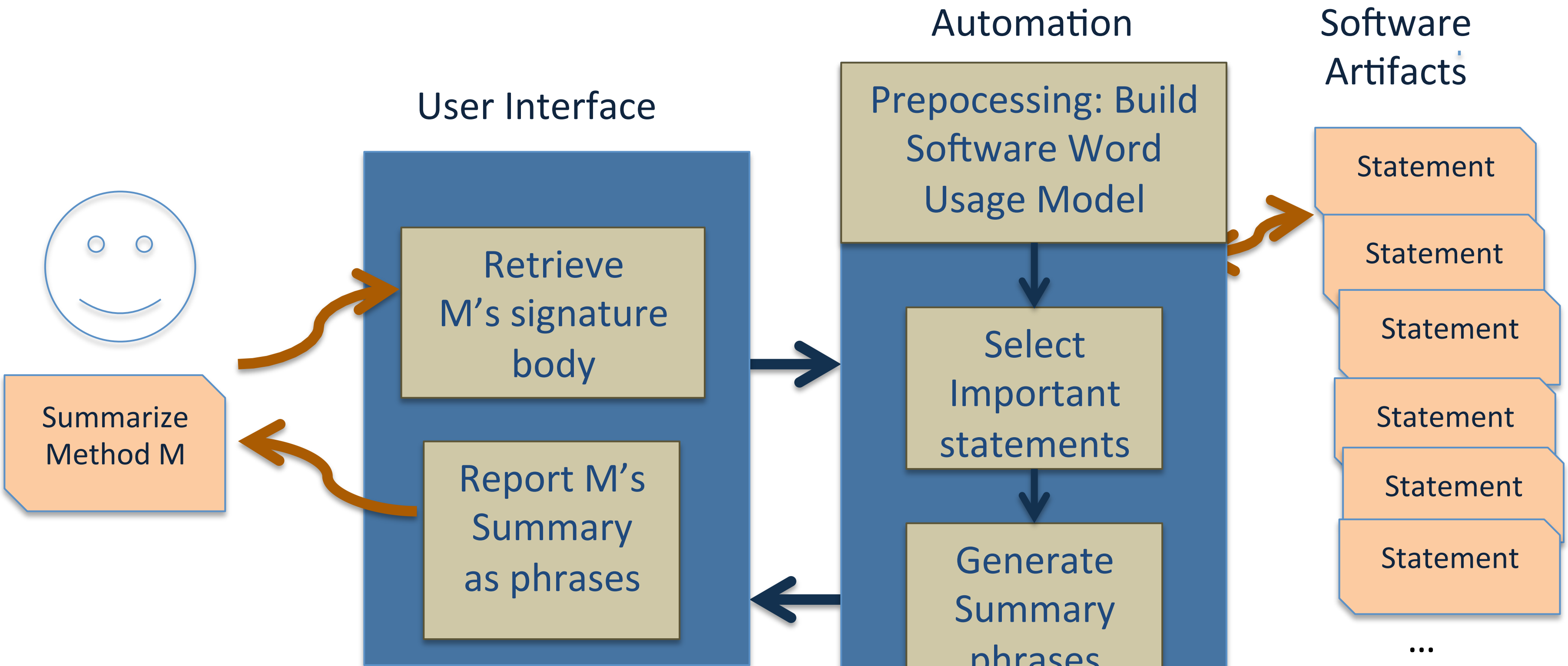
Method

Method

Method

…

Example: Method Summarization Tool



Summarize Method M

User Interface

Retrieve M’s signature body

Report M’s Summary as phrases

Automation

Prepocessing: Build Software Word Usage Model

Select Important statements

Generate Summary phrases

Software Artifacts

Statement

Statement

Statement

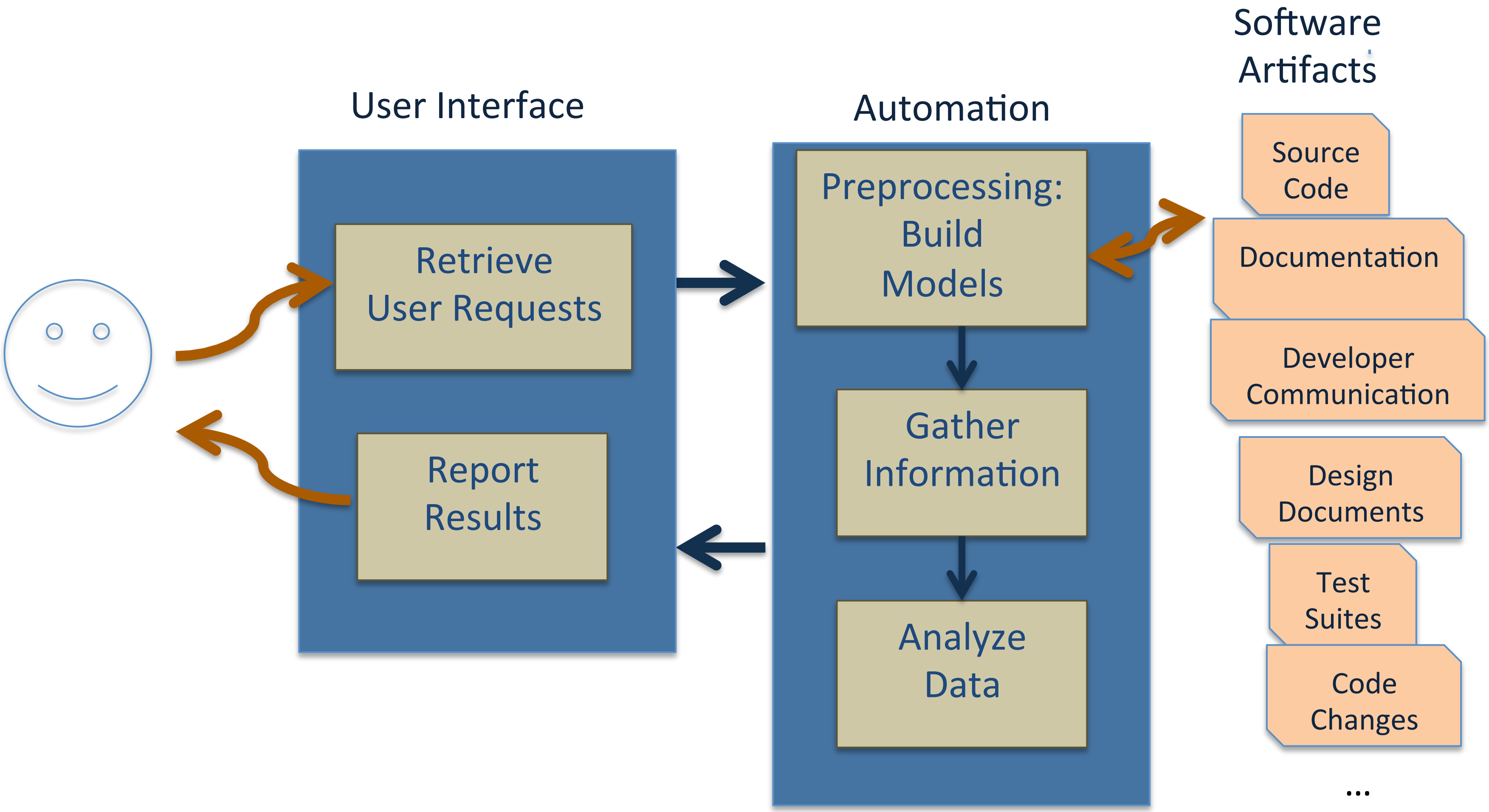
Statement

Statement

Statement

…

SE Power Tools Revisited



User Interface

Retrieve User Requests

Report Results

Automation

Preprocessing: Build Models

Gather Information

Analyze Data

Software Artifacts

Source Code

Documentation

Developer Communication

Design Documents

Test Suites

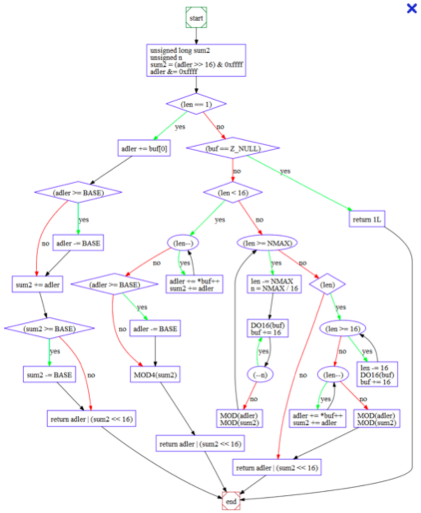
Code Changes

…

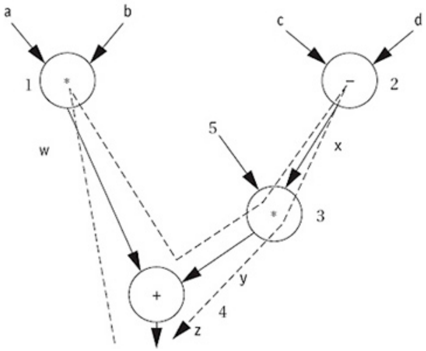
Power Tools:
What Information is used?

Structural:

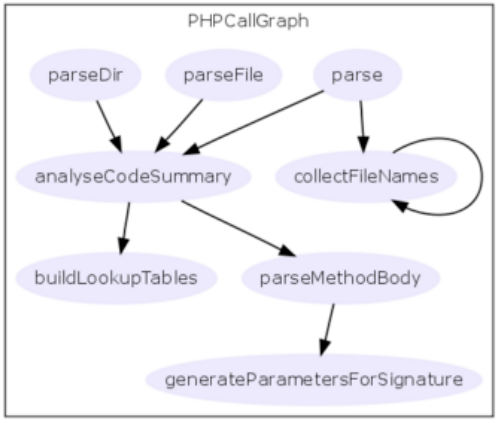
Control Flow



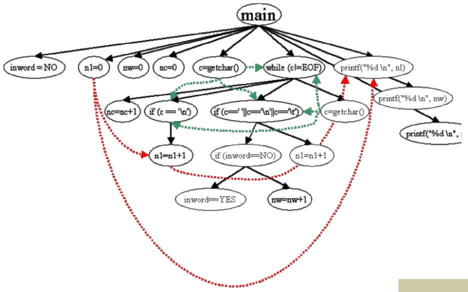
Data Flow



Call Graphs



Program Dependence



- Constants
- Types
- Inheritance

- Dynamic:
  - Frequency/order of execution
- Development Process-related:
  - Change logs, bug reports

What else is available?
Consider this code

```
public static int a(int c, int d) {
    int b;
    b = c * d;
    return b;
}
```

Compute and return a product

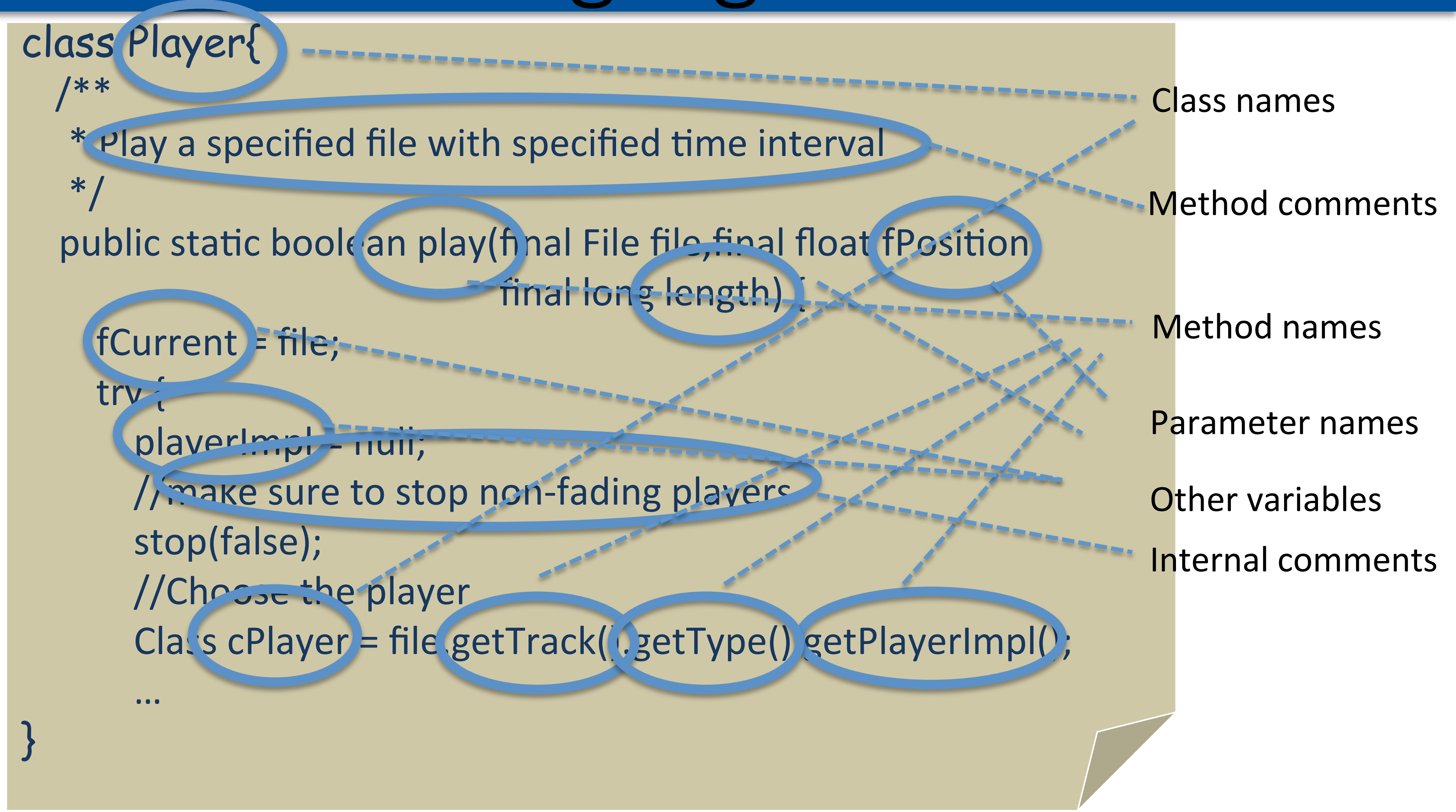
```
public static int c(int w, int h) {
    int a;
    a = w * h;
    return a;
}
```

Compute the area of a rectangle?

```
public static int computeArea(int width, int height) {
    int area;
    area = width * height;
    return area;
}
```

Given a width and height, compute & return the area of a rectangle, OBVIOUSLY.

Where is
Natural Language in Software?



```
class Player{
  /**
   * Play a specified file with specified time interval
   */
  public static boolean play(final File file,final float fPosition
                     final long length) {
    fCurrent = file;
    try {
      playerImpl = null;
      //make sure to stop non-fading players
      stop(false);
      //Choose the player
      Class cPlayer = file.getTrack().getType().getPlayerImpl();
      …
}
```

Class names

Method comments

Method names

Parameter names

Other variables

Internal comments

How can we leverage the naming?

```
class Player{
public static boolean play(final File file, final float fPosition,final long length) {
        fCurrent = file;
        try {
            playerImpl = null;
            stop(false);
      class cPlayer = file.getTrack().getType().getPlayerImpl();
        …}
```

Code Search

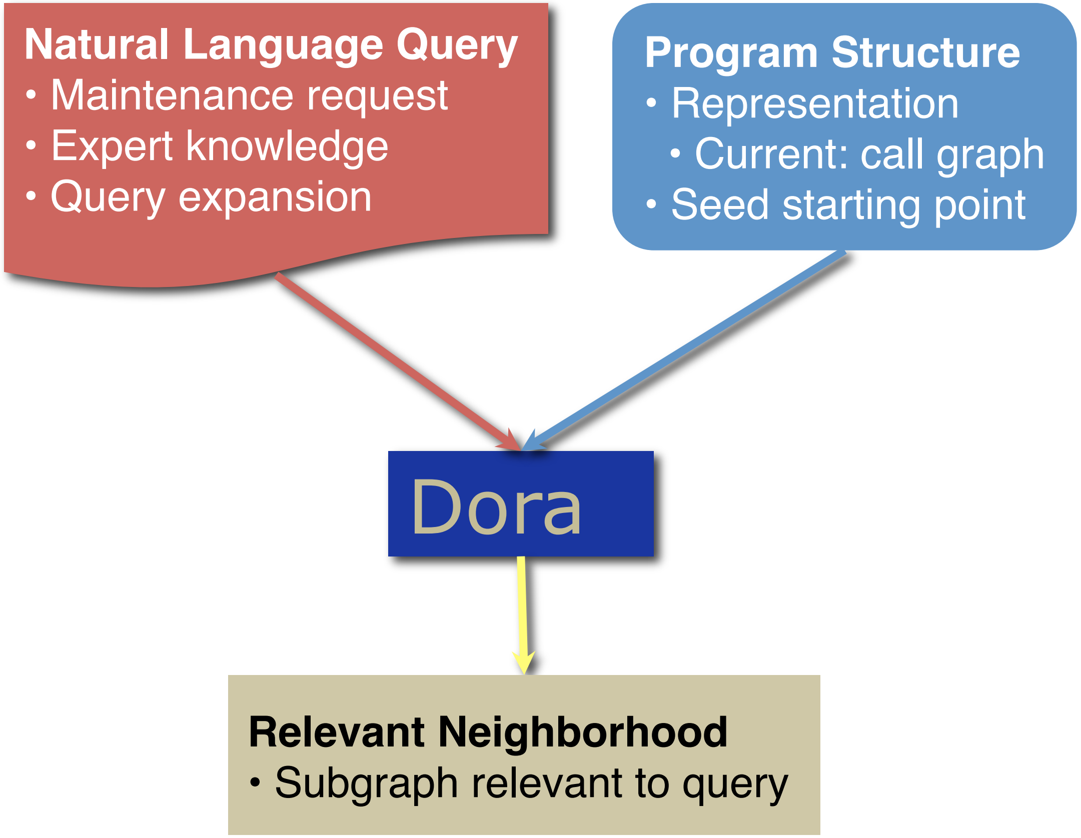
Traceability

Code Navigation

Refactoring

Marcus et al. study of literature revealed 25 different SE tasks!

Consider Dora the Program Explorer[*]

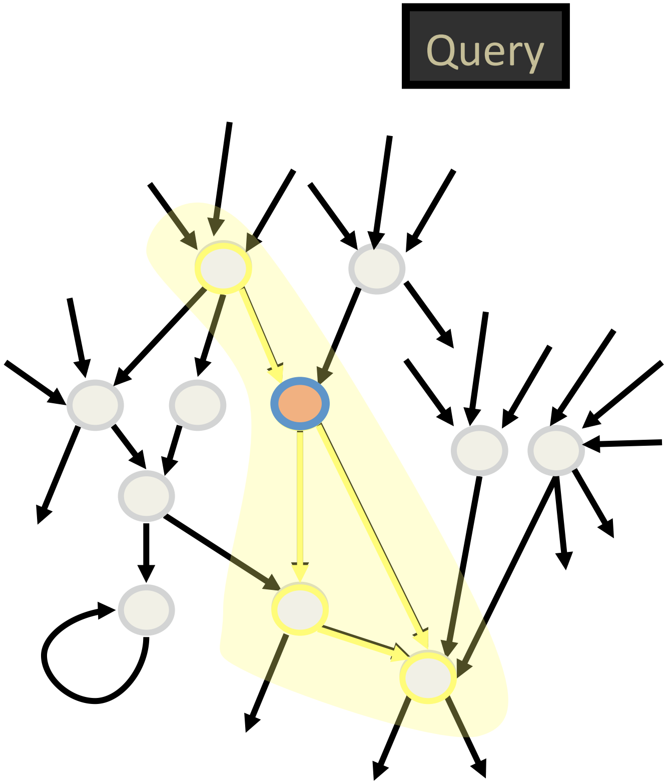


Natural Language Query
- Maintenance request
- Expert knowledge
- Query expansion

Program Structure
- Representation
  - Current: call graph
- Seed starting point

Dora

Relevant Neighborhood
- Subgraph relevant to query



Relevant
Neighborhood

[*] Dora comes from *exploradora*, the Spanish word for a female explorer. Hill et al.

Maintenance Scenario

Program: JBidWatcher, eBay auction sniping program

Bug: When a user triggers an add auction,
nothing happens – there is no effect.

SE Task: Locate code related to 'add auction' trigger

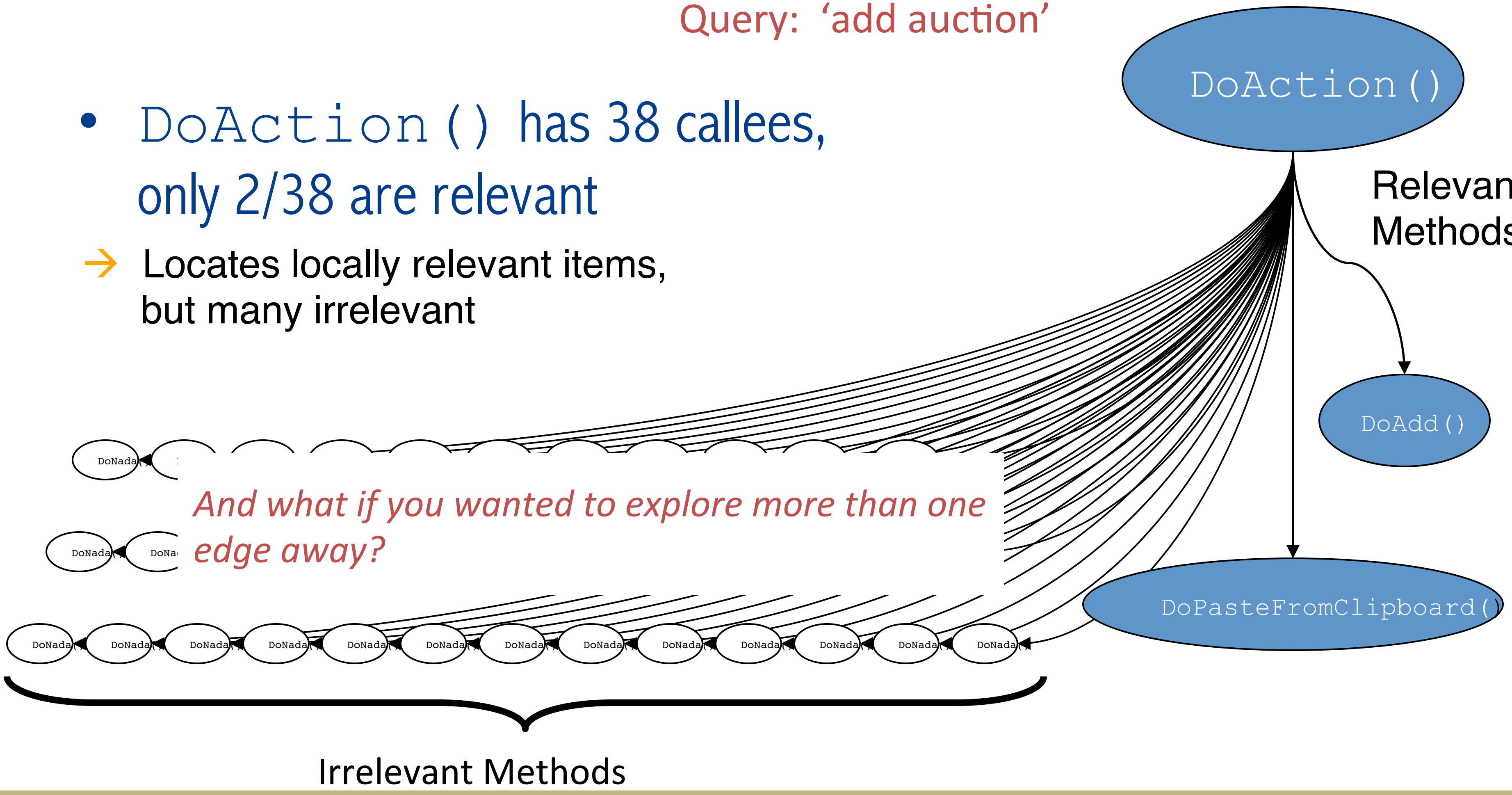
Seed: `DoAction()` method, from prior knowledge

Using only structural information

Query: 'add auction'

- DoAction() has 38 callees, only 2/38 are relevant

→ Locates locally relevant items, but many irrelevant



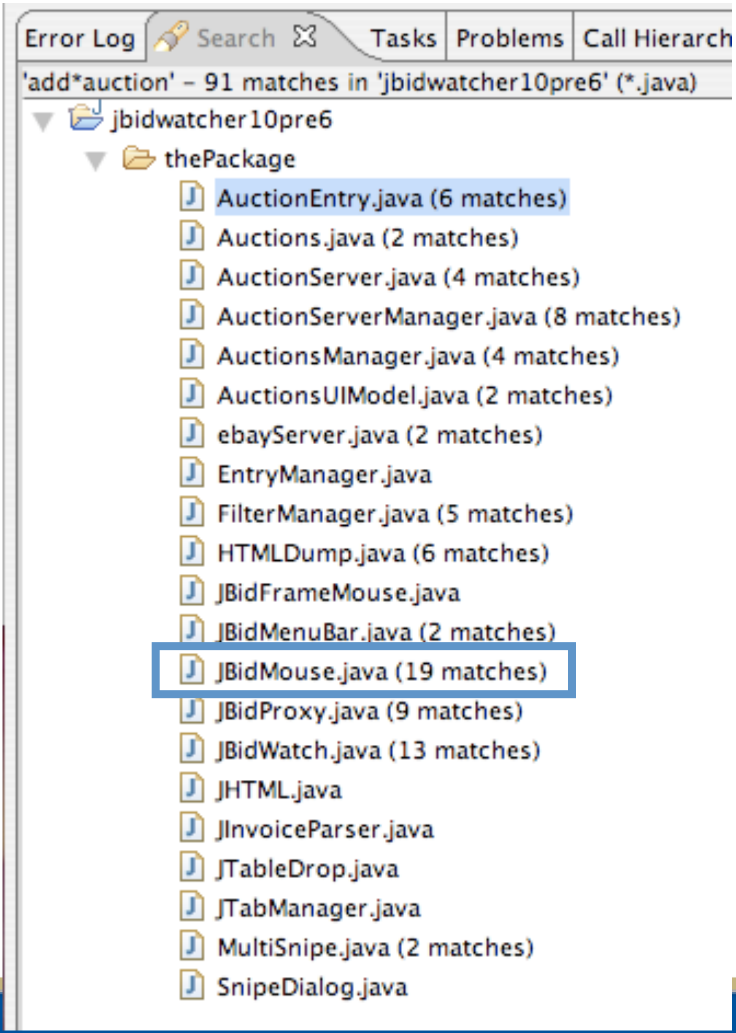
And what if you wanted to explore more than one edge away?

Using only lexical information

- 50/1812 methods contain matches to ‘add*auction’ regular expression query

- Only 2/50 are relevant

→Locates globally relevant items, but many irrelevant



'add*auction' – 91 matches in 'jbidwatcher10pre6' (*.java)

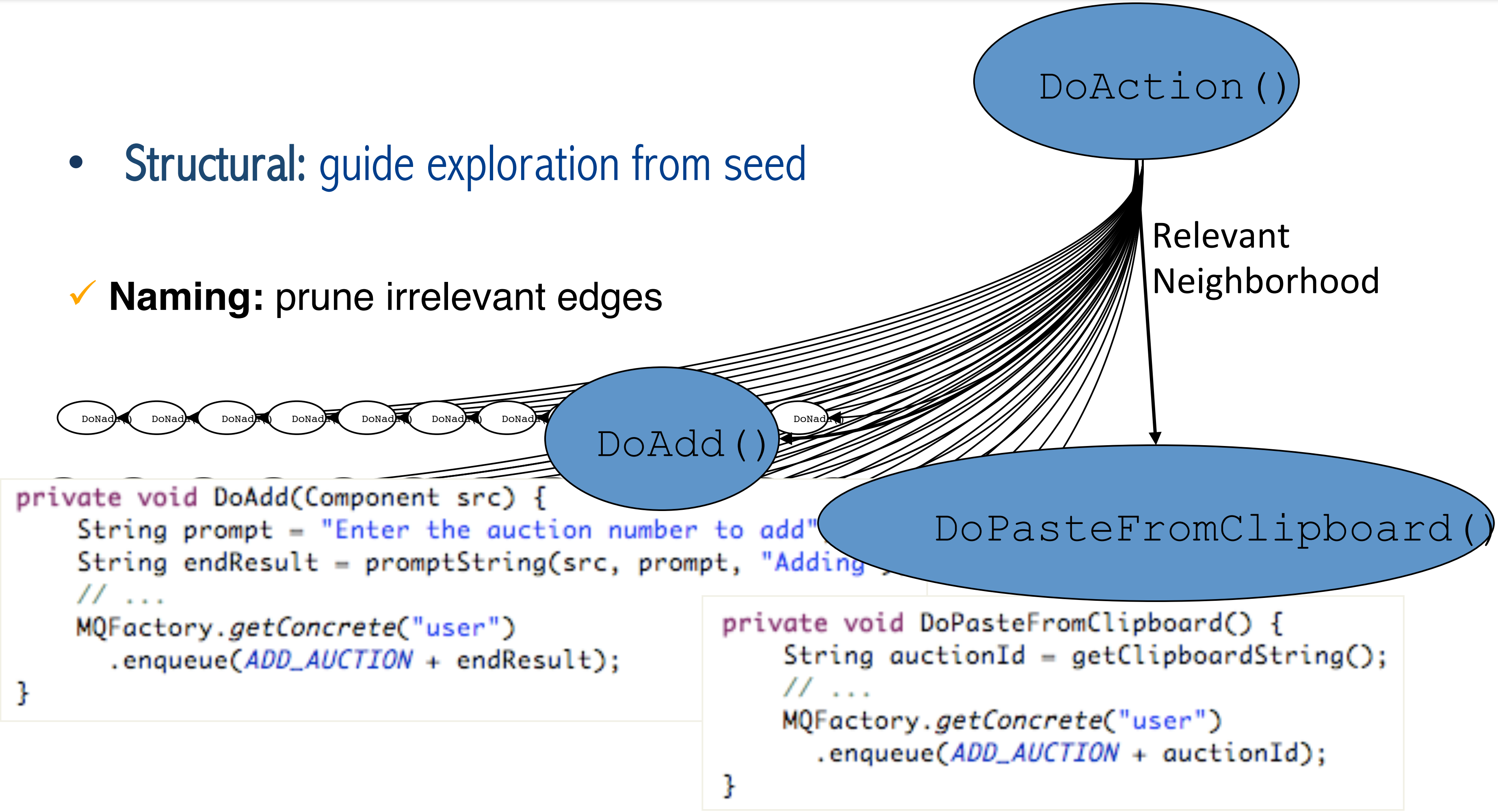
- jbidwatcher10pre6
  - thePackage
    - AuctionEntry.java (6 matches)
    - Auctions.java (2 matches)
    - AuctionServer.java (4 matches)
    - AuctionServerManager.java (8 matches)
    - AuctionsManager.java (4 matches)
    - AuctionsUIModel.java (2 matches)
    - ebayServer.java (2 matches)
    - EntryManager.java
    - FilterManager.java (5 matches)
    - HTMLDump.java (6 matches)
    - JBidFrameMouse.java
    - JBidMenuBar.java (2 matches)
    - JBidMouse.java (19 matches)
    - JBidProxy.java (9 matches)
    - JBidWatch.java (13 matches)
    - JHTML.java
    - JInvoiceParser.java
    - JTableDrop.java
    - JTabManager.java
    - MultiSnipe.java (2 matches)
    - SnipeDialog.java

Combining structural & lexical

Query:
'add auction'

- **Structural:** guide exploration from seed

✓ **Naming:** prune irrelevant edges



```
private void DoAdd(Component src) {
    String prompt = "Enter the auction number to add"
    String endResult = promptString(src, prompt, "Adding
    // ...
    MQFactory.getConcrete("user")
      .enqueue(ADD_AUCTION + endResult);
}
```

```
private void DoPasteFromClipboard() {
    String auctionId = getClipboardString();
    // ...
    MQFactory.getConcrete("user")
      .enqueue(ADD_AUCTION + auctionId);
}
```

Text Analysis in SE

Challenges

Achievements

Opportunities

So, what is Text Analysis?

analysis of the natural language used by programmers in writing software

(source code + other software artifacts)

Why?

To provide important information
for building automated and semi-automated
recommendation systems and analysis tools
to support SE tasks

Flavors of Text Analysis

Information/Text Retrieval (IR/TR)

Given query words, retrieve documents containing unstructured data related to those topics:

* For a known information need, return as many relevant docs as possible
* To enable the user to explore a problem domain

Natural Language Processing (NLP)

Software that will automatically analyze, understand, and generate languages that humans use naturally (e.g., English)

* To know what concepts a word or phrase represents
* To know how to link those concepts together in a meaningful way

In Source code: Comments and Identifiers

Natural Language in Comments: Different Types (by content)

- Descriptive

  /* show save dialog and get file name */

- Notes

  /* TODO: fix this! */

- Cross-reference

  /* @see setData */

- Explanatory

  /* we clone the vector to avoid deadlock */

- And other types ….

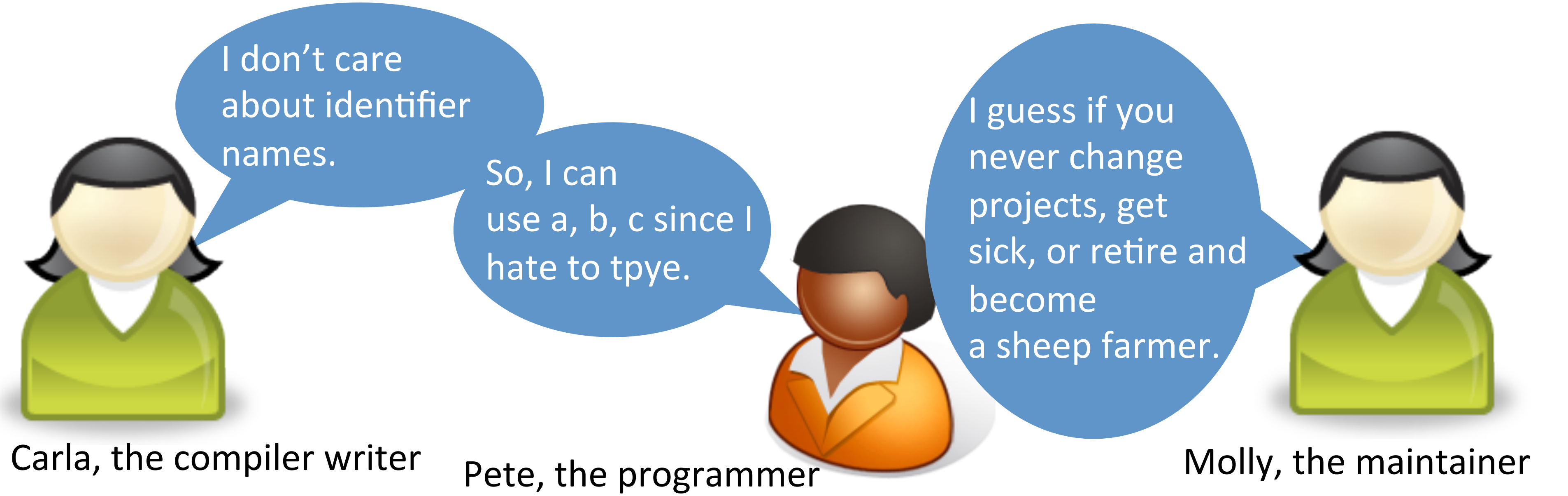
Natural Language in Descriptive Comments: Conventions

```
// Play a specified file with specified time interval
```

```
/*  Registers the text to display in a tool tip. The text  displays when the
*  cursor lingers over the component.
* @param text the string to display. If the text is null,  the tool tip is
*  turned off for this component. */
```

- Not a full sentence
- Multiline -> later, full sentences with period
- 1st line: Often starts with a verb and then the direct object
- Contain Java doc components

Natural Language in Identifiers: Significance & Studies


I don't care about identifier names.
So, I can use a, b, c since I hate to tpye.
I guess if you never change projects, get sick, or retire and become a sheep farmer.
Carla, the compiler writer
Pete, the programmer
Molly, the maintainer

Identifiers play a key role in program comprehension and follow conventions.

- Useful for software tools: metrics, traceability, program understanding
- Metaphors, morphology, scope, part of speech hints

- [Caprile &Tonella] [Liblit et al.] [Deissenboeck & Pizka], Lawrie, Binkley et al.] [Host & Oestvold]

Natural Language in Identifiers Conventions

month average_score medianScore cWord2Num

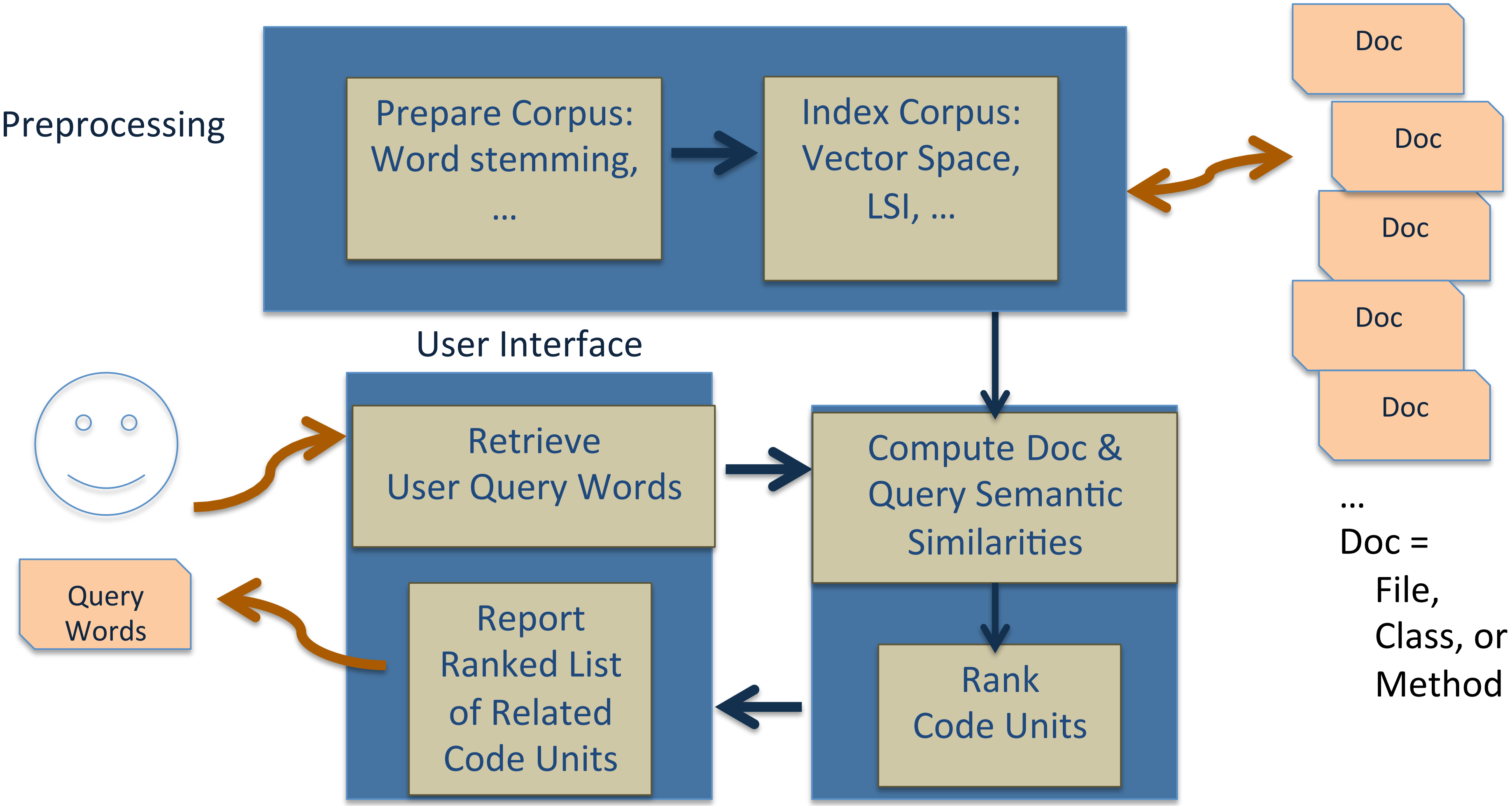
hostname sizeof SIMPLETYPENAME

jLabel PHP_id cmp
ASTVisitorTree ConvertASCIItoUTF

sortList sortedList

- Single and multiple words (multi-words)
- Camel case and underscores for visible split, but not always
- Abbreviations, sometimes different semantics in different code units
- Conventions based on entity being named

Text Retrieval: Overview



Preprocessing

Prepare Corpus: Word stemming, …

Index Corpus: Vector Space, LSI, …

User Interface

Retrieve User Query Words

Compute Doc & Query Semantic Similarities

Rank Code Units

Report Ranked List of Related Code Units

Query Words

Doc

Doc

Doc

Doc

Doc

…

Doc = File, Class, or Method

Text Retrieval in SE: Example

```
class Player{
public static boolean play(final File file,final float fPosition,final long length) {
        fCurrent = file;
        try {
            playerImpl = null;
            stop(false);
      class cPlayer = file.getTrack().getType().getPlayerImpl();
       …}
```

Prepare Corpus: Remove non-literals/stop words; Split ids; Stem

```
Play  play  file  f Position  length
        f Current  file
        play Impl   stop
      c Play  file   get Track   get Type    get Play Impl
```

Text Retrieval in SE: Example

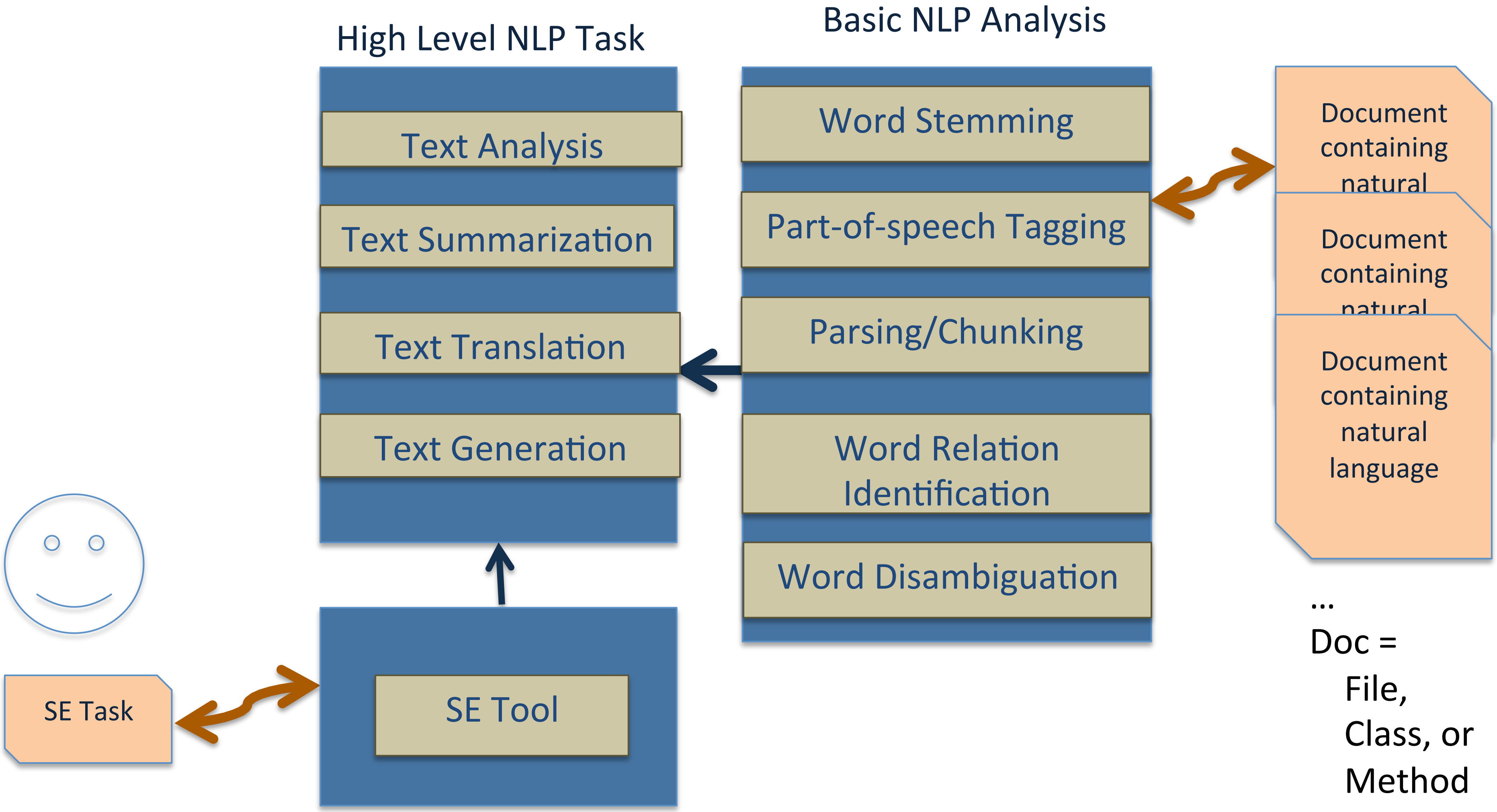
Play play file f Position length
f Current file
play Impl stop
c Play file get Track get Type get Play Impl

Index Corpus

	play	file	f	position	length	current	impl	stop	c	get	track	type	…
C1	5	3	2	1	1	1	2	1	1	3	1	1	
C2	…	…	…	…	…	…	…	…	…	…	…	…	

Process query against indexed corpus -> ranked list of relevant docs

NLP: Overview



High Level NLP Task

Text Analysis

Text Summarization

Text Translation

Text Generation

Basic NLP Analysis

Word Stemming

Part-of-speech Tagging

Parsing/Chunking

Word Relation Identification

Word Disambiguation

Document containing natural

Document containing natural

Document containing natural language

SE Task

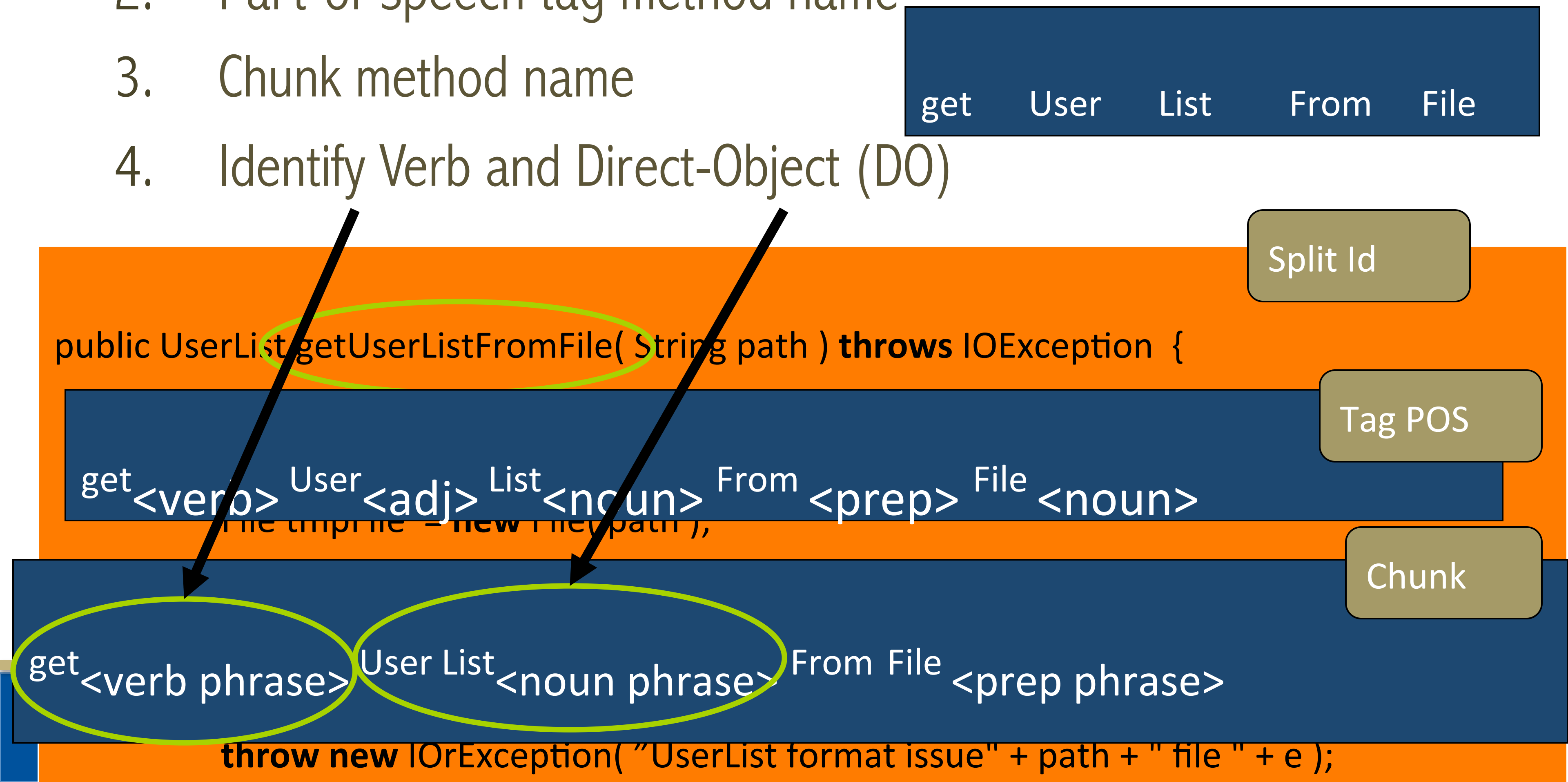
SE Tool

…

Doc =
File,
Class, or
Method

NLP in SE: An Example

1. Split Name into Words
2. Part-of-speech tag method name
3. Chunk method name
4. Identify Verb and Direct-Object (DO)



get User List From File

Split Id

```
public UserList getUserListFromFile( String path ) throws IOException {
        File tmpFile = new File( path );
        throw new IOrException( "UserList format issue" + path + " file " + e );
```

Tag POS

get <verb> User <adj> List <noun> From <prep> File <noun>

Chunk

get <verb phrase> User List <noun phrase> From File <prep phrase>

NLP in SE:
Generating Phrases by Lexicalization

```
print(current);
```

/* print current */ But what is *current?*

/* print current document */ ✓

Context implies what *'current'* is;
→ Type information can provide context

Type Name	Variable Name	Generated Phrase
CallFrame	parentFrame	**parent call frame**

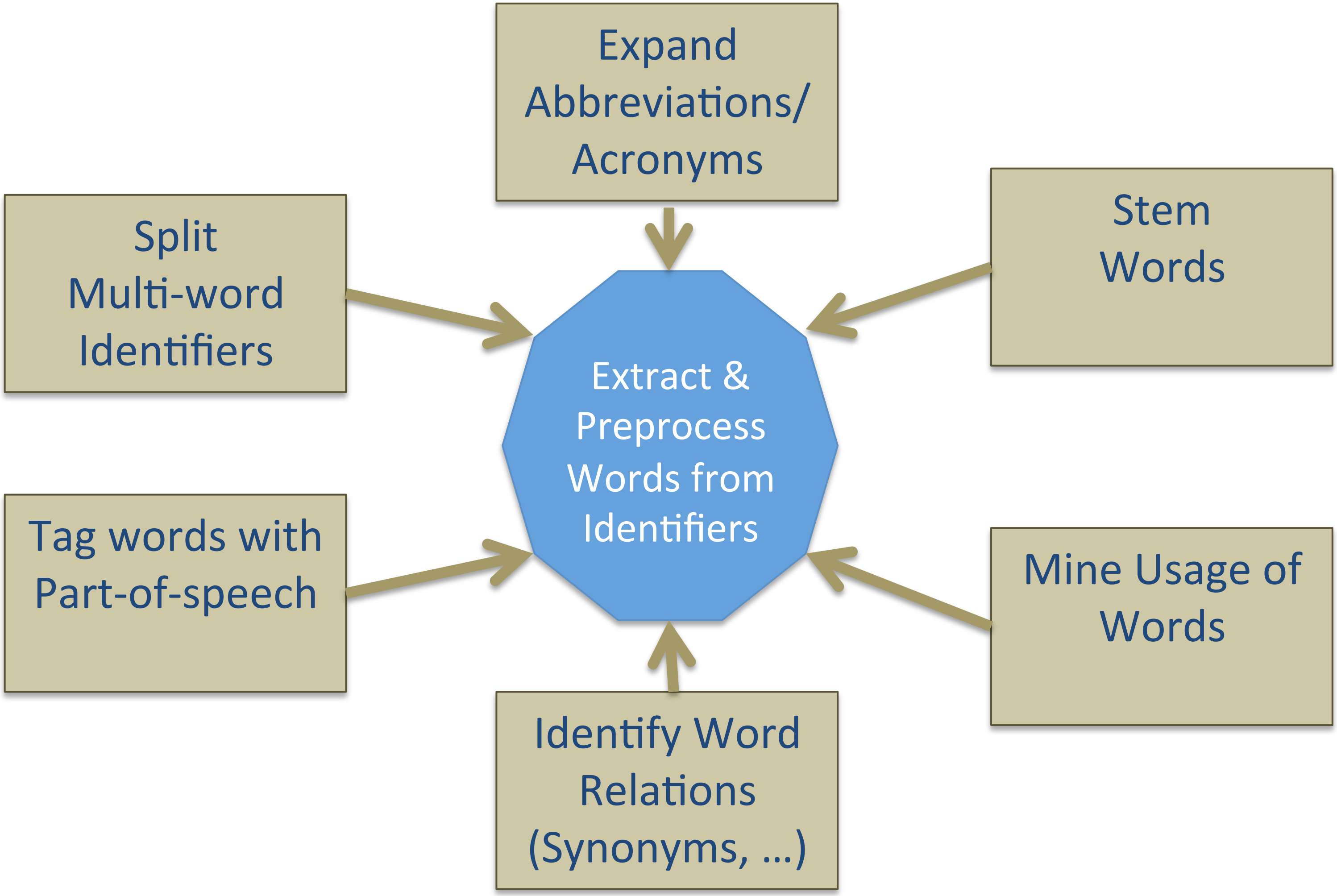
Selectable is an adjective

Frame is not repeated

- **Phrase Generation leverages:**
  - Part of speech of words in type and variable names
  - Overlap between type & variable names

Sridhara et al.

For both TR and NLP in SE: Lexical-level Analysis



Expand Abbreviations/ Acronyms

Split Multi-word Identifiers

Stem Words

Extract & Preprocess Words from Identifiers

Tag words with Part-of-speech

Mine Usage of Words

Identify Word Relations (Synonyms, …)

Splitting Multi-words

Challenges

- Mixed case: medianScore
- Same case: sortedList, notype, textbox
- Abbreviations: ASTVisitorNode, cmp

Current Strategies

- Standard & customized dictionaries
- Word frequencies in code
- Abbreviation expansion during id splitting

None have conquered the same case problem to high accuracy.

Expanding Abbreviations

Challenges (of nondictionary words)

Prefix (attr, obj, param, i);
Acronyms (ftp)
Combination (println)

Dropped Letter (msg)
Misspelling (instanciation)
No boundary (filesize)

And, the same abbreviation can have different expansions depending on domain or context

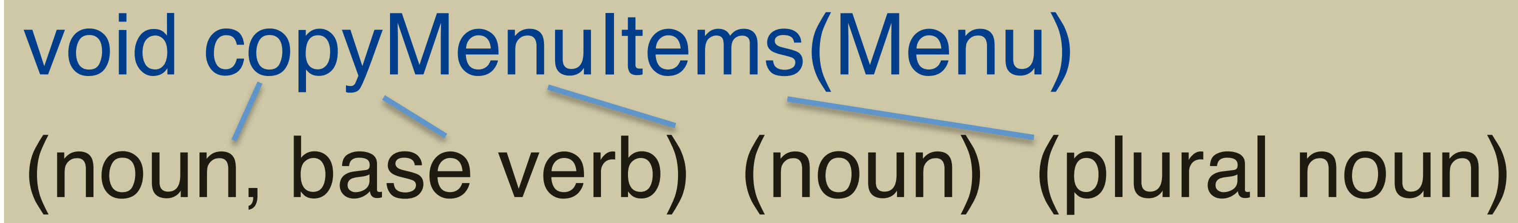
inst	**CFG**
Instance Instruction Instantiate Install ???	Control Flow Graph Context-Free Grammar Configuration Configure ??? uh oh

Current Strategies

- Manually create table of common short forms in code
- Mine expansions from the code, look nearby first

Tagging Part of Speech

Challenges



void copyMenuItems(Menu)

(noun, base verb) (noun) (plural noun)

Boolean copiedItem()

Current Strategies

Develop rules based on
naming conventions,
entity being named,
context of entity

Solving Vocabulary Mismatch: Identifying Word Relations

What words are similar to “remove”?

Remove ← **Delete/Withdraw/Eliminate**

- Humans: Refine query by adding related words
  - Error prone and time consuming

Strategies

Some IR techniques can automatically expand query:

•Digital thesaurus with semantic similarity

•Latent Semantic Indexing and related approaches

Synonyms are not always enough for searching

Query : "money transaction"   Not successful
Query : "bank transaction"   successful
But <money, bank> not synonymous

Other Semantic Similarity Types:

Hypernyms and

Words with general/specific meaning

All these types can be identified by current semantic similarity techniques. (WordNet)
But not always adequate for software.

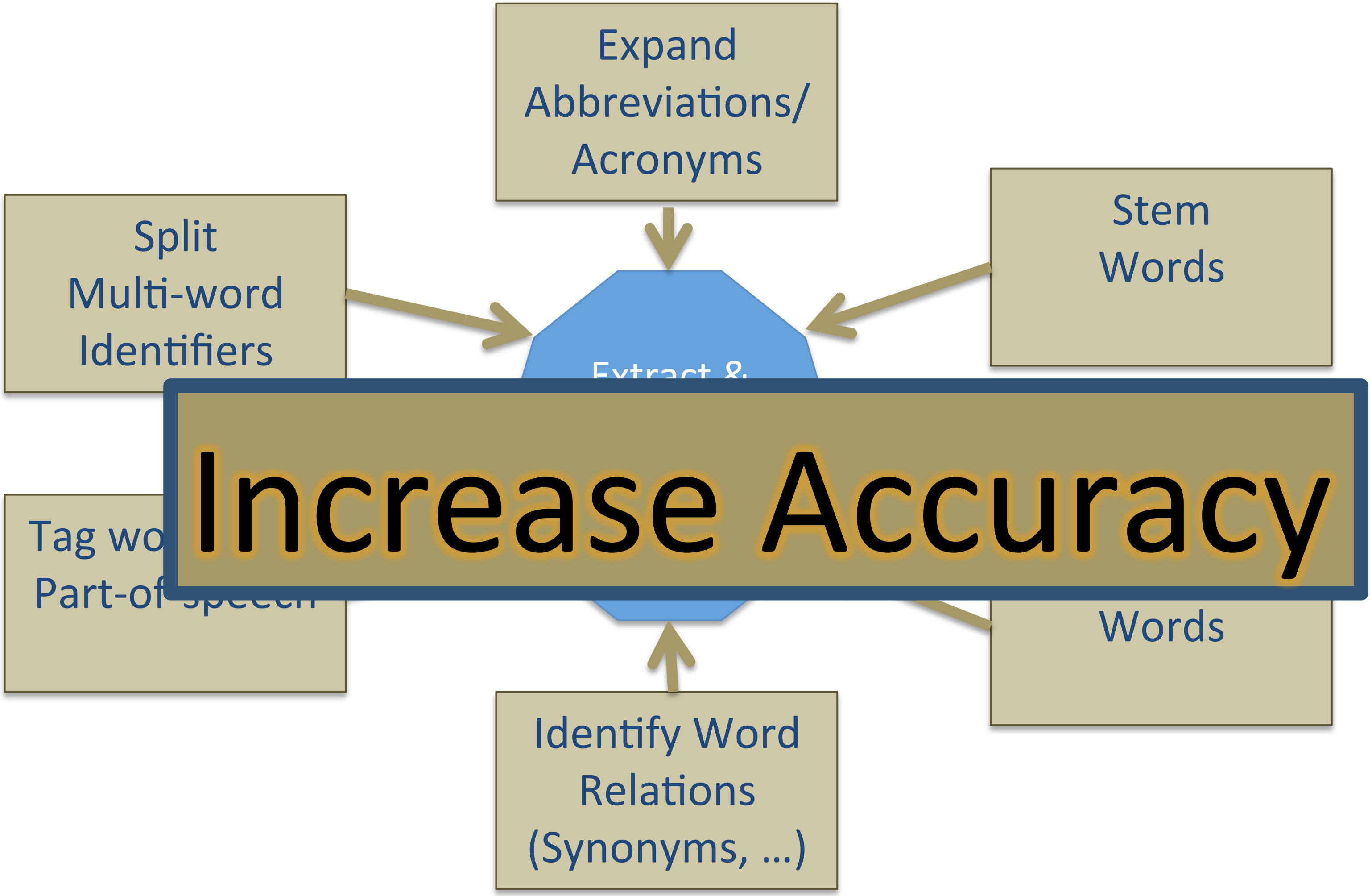
Meronyms and Holonyms

Wheel → meronym of Car (part-of)

Topically related

Words belonging to the same topic
Bank, Check, Money, Deposit

Lexical-level Analysis Opportunities



Expand Abbreviations/ Acronyms

Split Multi-word Identifiers

Stem Words

Extract &

Tag wo

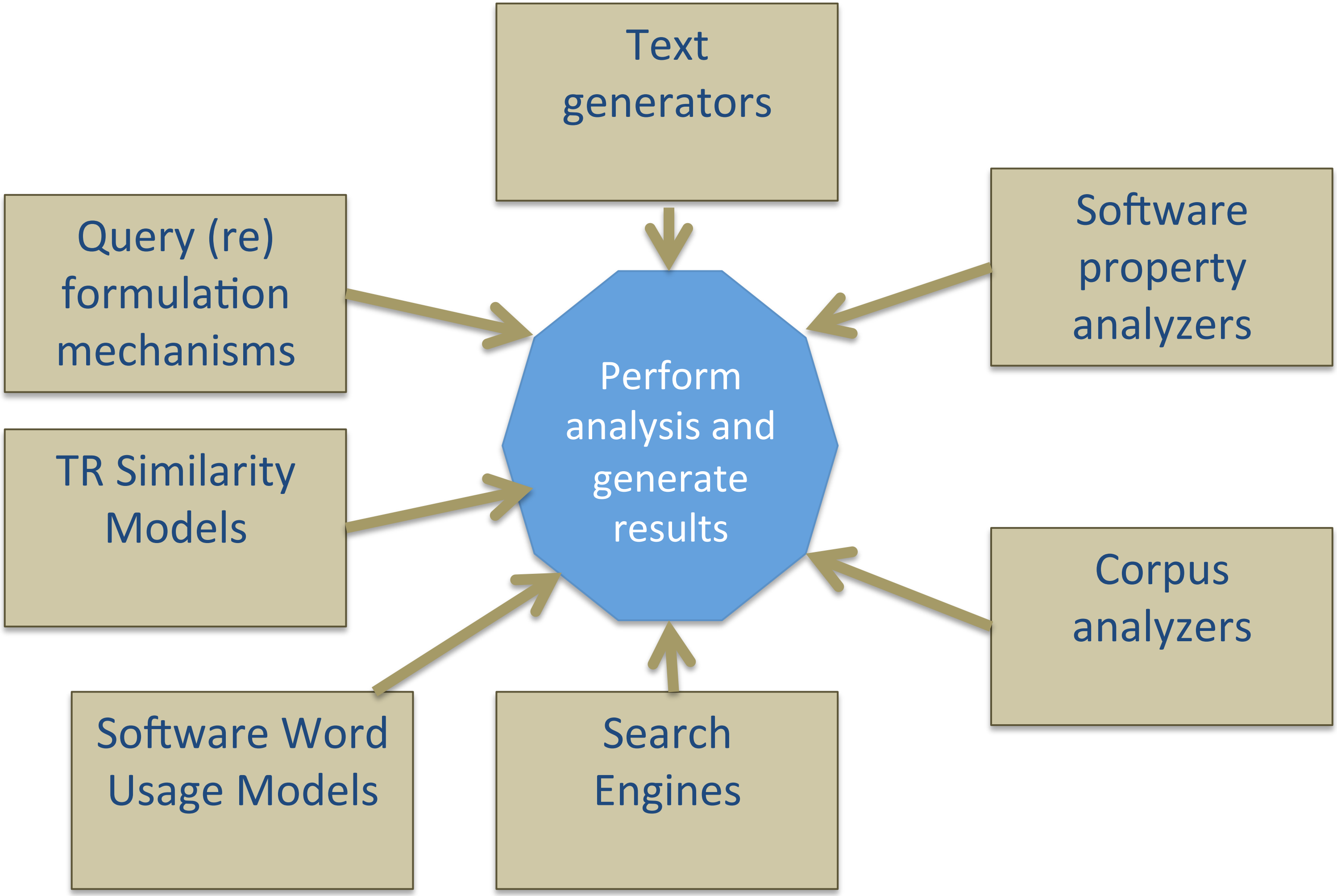
Part-of speech

Words

Identify Word Relations (Synonyms, …)

Increase Accuracy

Corpus-level Analysis



Text generators

Query (re) formulation mechanisms

Software property analyzers

Perform analysis and generate results

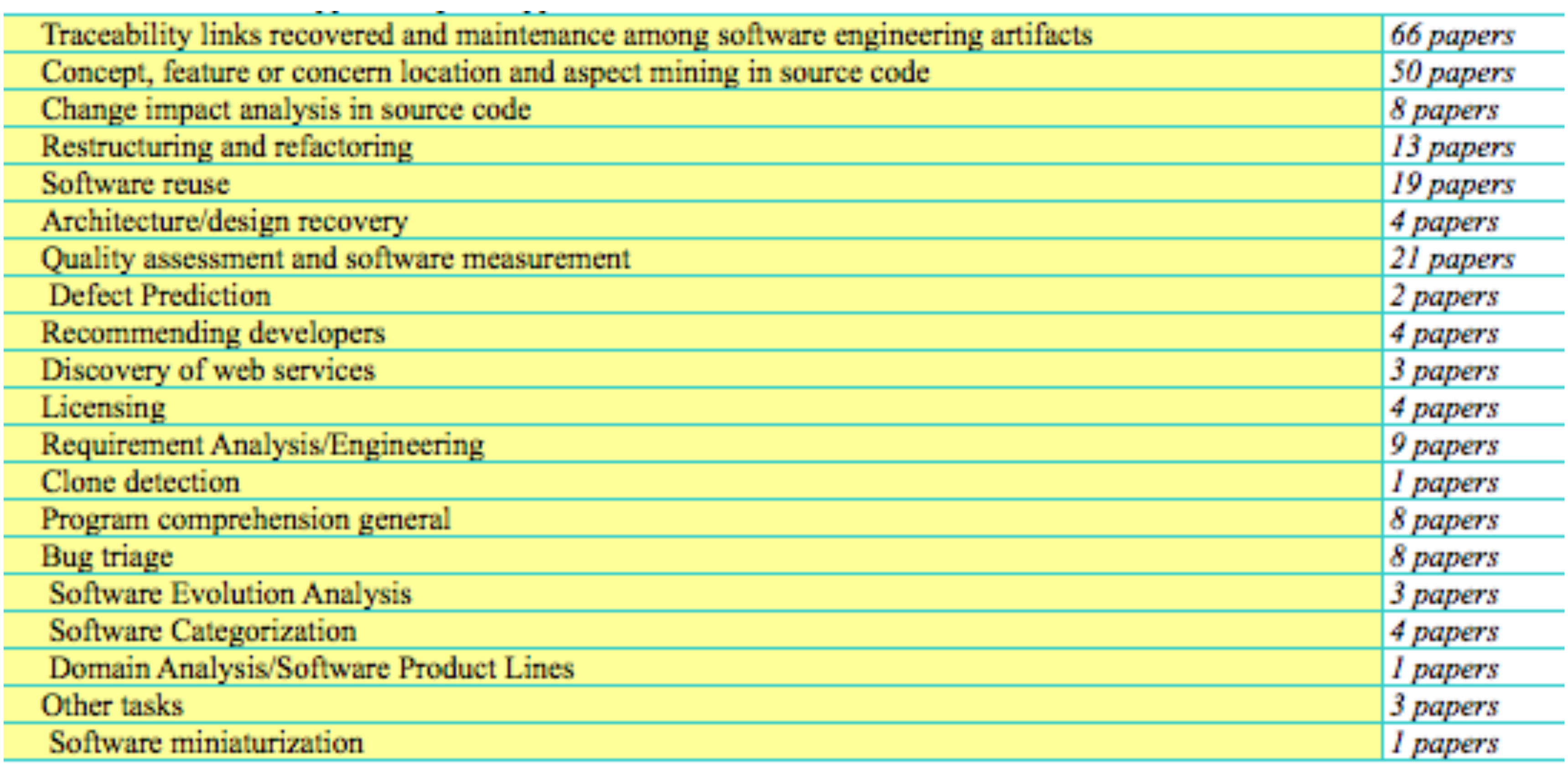
TR Similarity Models

Corpus analyzers

Software Word Usage Models

Search Engines

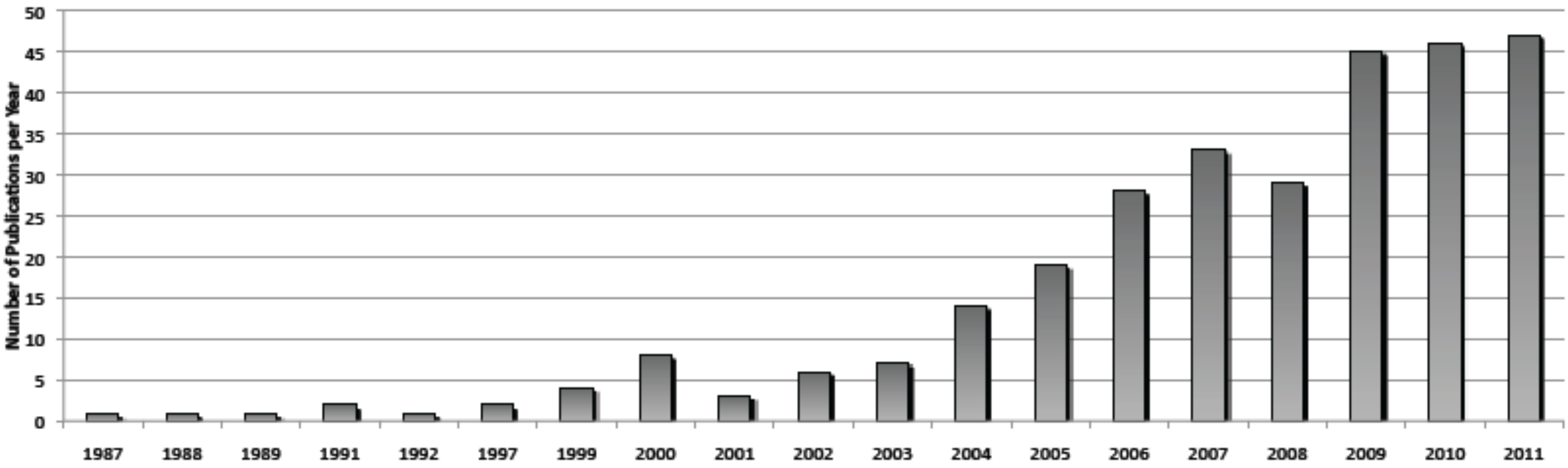
Many Uses of Text Analysis



Traceability links recovered and maintenance among software engineering artifacts	*66 papers*
Concept, feature or concern location and aspect mining in source code	*50 papers*
Change impact analysis in source code	*8 papers*
Restructuring and refactoring	*13 papers*
Software reuse	*19 papers*
Architecture/design recovery	*4 papers*
Quality assessment and software measurement	*21 papers*
Defect Prediction	*2 papers*
Recommending developers	*4 papers*
Discovery of web services	*3 papers*
Licensing	*4 papers*
Requirement Analysis/Engineering	*9 papers*
Clone detection	*1 papers*
Program comprehension general	*8 papers*
Bug triage	*8 papers*
Software Evolution Analysis	*3 papers*
Software Categorization	*4 papers*
Domain Analysis/Software Product Lines	*1 papers*
Other tasks	*3 papers*
Software miniaturization	*1 papers*

Marcus et al.

Growth of Text Analysis



Marcus et al.

Going Forward
with Text Analysis

* ***Apply*** text analysis to

- develop new tools and improve tools

* ***Combine*** information

- Structure + Text + Dynamic

**** Explore configurations*** of analyses

* ***Improve Evaluations***

- Lack of common infrastructure

Source Code

Documentation

Developer Communication

Design Documents

Test Suites

Code Changes

Participate At ICSM 2012

The Next Five Years of Text Analysis in Software Maintenance

TODAY: 15:35 – 17:35

Belvedere