API refactoring using natural language and program synthesis

Yanxin Lu, Rice University
Swarat Chaudhuri, Rice University
Christopher Jermaine, Rice University

RICE

Software refactoring

- Library/platform upgrade
- Obsolete code reuse

```
SSHClient ssh = new SSHClient();
SFTPClient ftp = ssh.newSFTPClient();
ssh.authPassword(username, password);
ssh.connect(host);
ftp.ls(path);
ftp.close();
```

API refactoring

```
FTPClient f = new FTPClient();
f.connect(host);
f.login(username, password);
FTPFile[] files = f.listFiles(path);
f.disconnect();
```

Almost as hard as coding

Problem

Can we automate the process of API refactoring using program synthesis?

Contribution

- Combination of two existing techniques
  - API translation
    - Natural language
  - API sequence synthesis
    - Complete API sequence
    - Bayou

Related Work

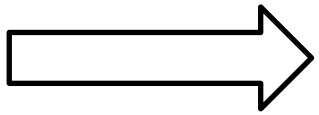
- API mapping
  - Natural language
  - Sequence alignment
- API sequence synthesis
  - Learning from the web (SWIM)
  - Bayou

Raghothaman, Mukund, Yi Wei, and Youssef Hamadi. "SWIM: Synthesizing What I Mean-Code Search and Idiomatic Snippet Synthesis." *Software Engineering (ICSE), 2016 IEEE/ACM 38th International Conference on*. IEEE, 2016.
Murali, Vijayaraghavan, et al. "Neural Sketch Learning for Conditional Program Generation." *arXiv preprint arXiv: 1703.05698* (2017).

Algorithm

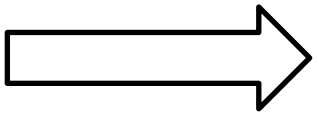
- A()
- B()
- C()
- D()
- E()

API translation



- a()
- b()
- c()
- d()
- e()

API synthesis

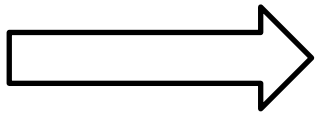


- a()
- b()
- c()
- d()
- e()

Algorithm

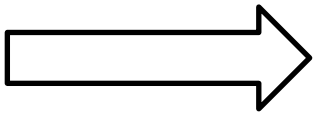
- A()
- B()
- C()
- D()
- E()

API translation



- a()
- b()
- c()
- d()
- e()

API synthesis

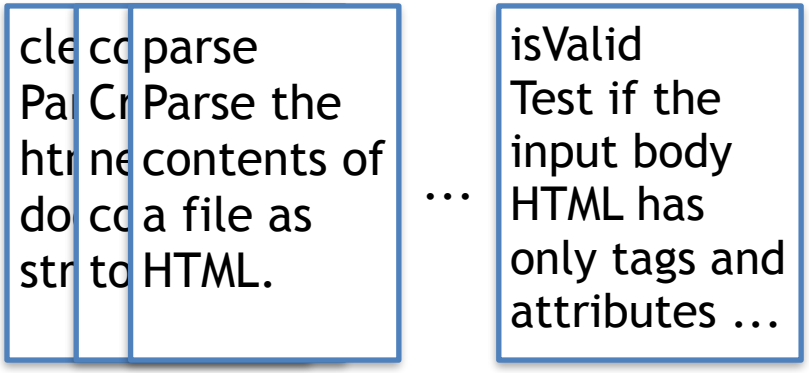


- a()
- b()
- c()
- d()
- e()

API translation

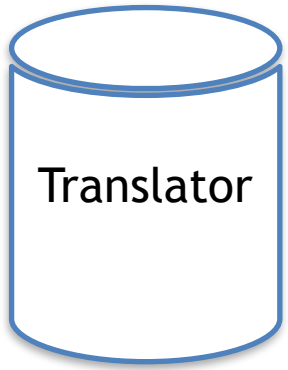
All relevant libraries and Java 8

Text extraction



Train a word2vec model

- A()
- B()
- C()
- D()
- E()



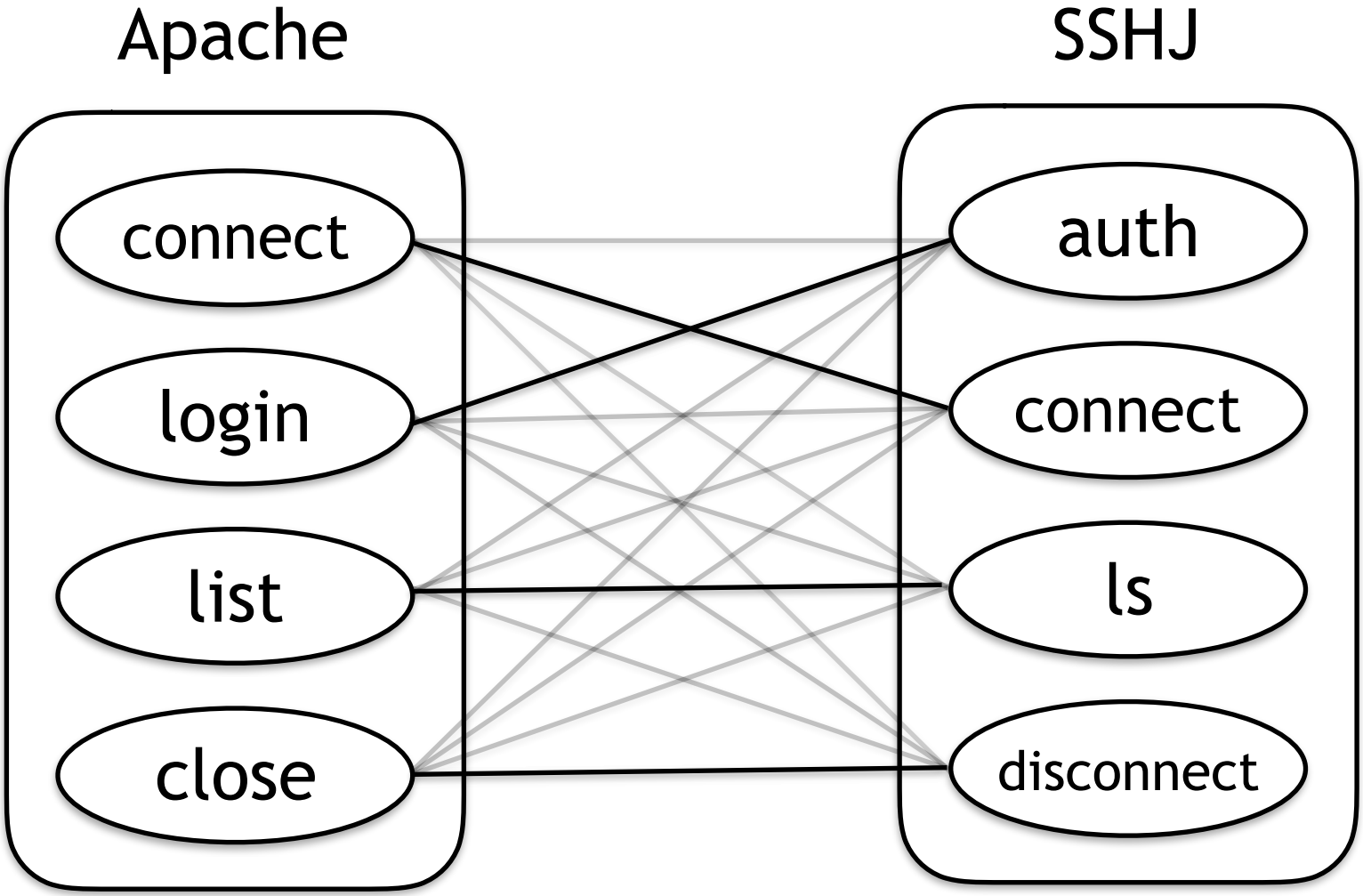
- a()
- b()
- c()
- d()
- e()

Word2Vec model

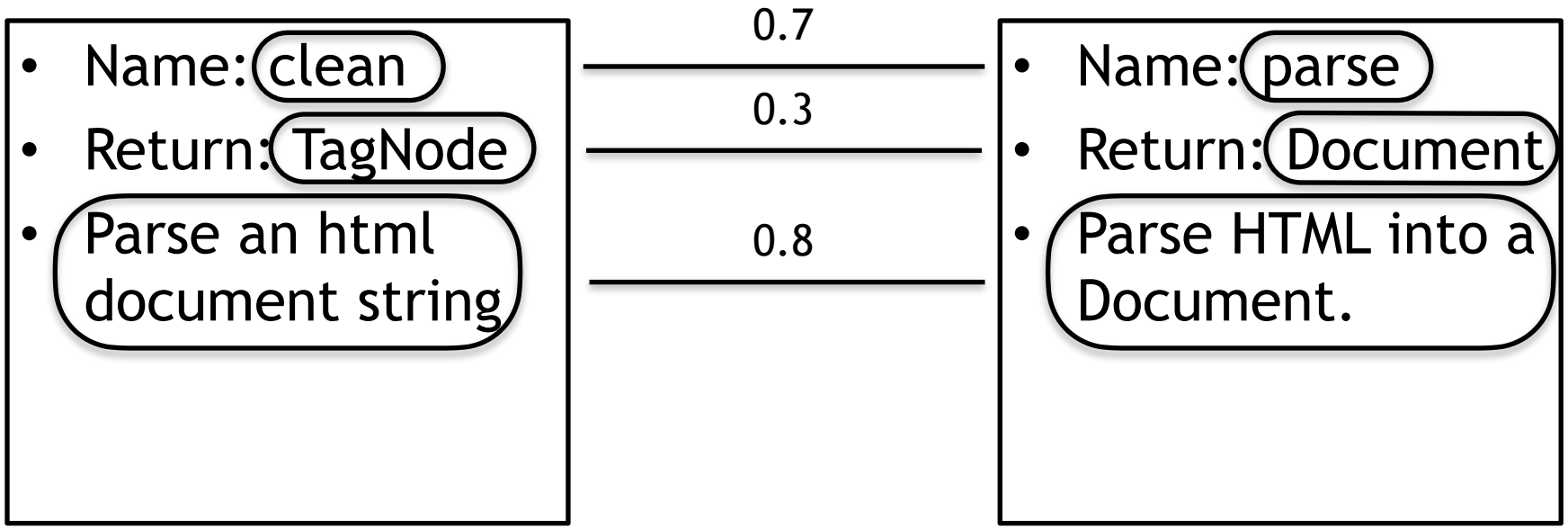
- Captures some degree of semantic information

Query word	Similar words
int	integer, float, long, double, short
ftp	nntp, smtp, secret, pixmap, out-of-synch
button	rollover, radio, tooltip, checkbox, click
index	IndexFrom, MenuIndex, ListIndex, occurrence, nth
stream	InputStream, StreamB, BufferTest, console, AccessFile
image	gif, animation, texture, BufferedImage, RenderedImage
email	bcc, recipient, sender, addresse, mail
vector	scalar, dense, product, kernel, matrix

Pair-wise API similarities



API similarity

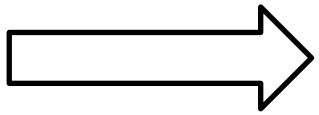


- Similarity = w1 * 0.5 + w2 * 0.3 + w3 * 0.8
- More weight on description similarity
- Word list similarity - bipartite matching

Algorithm

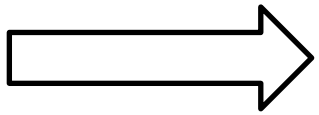
- A()
- B()
- C()
- D()
- E()

API translation



- a()
- b()
- c()
- d()
- e()

API synthesis



- a()
- b()
- c()
- d()
- e()

API sequence synthesis

- Input: APIs in the target library
  - Stand-alone APIs
  - Might miss a few APIs
- Output: a complete API call sequence
- Bayou

Murali, Vijayaraghavan, et al. "Neural Sketch Learning for Conditional Program Generation." *arXiv preprint arXiv: 1703.05698* (2017).

API sequence synthesis

- Input: APIs in the target library
  - Stand-alone APIs
  - Might miss a few APIs
- Output: a complete A
- Bayou

```
void read_href(String content,
               String selector,
               String attr,
               Evaluator _arg01) {
  ///call:parse type:Jsoup call:select
  ///call:first
}
```

Evidence: types, calls

```
void read_href(String content,
               String selector,
               String attr,
               Evaluator _arg01) {
  Elements e1;
  Document d1;
  Elements e3;
  Element e2;
  d1 = Jsoup.parse(content);
  e1 = d1.select(selector);
  e2 = e1.first();
  e3 = Collector.collect(_arg01, e2);
}
```

Bayou

Code corpus

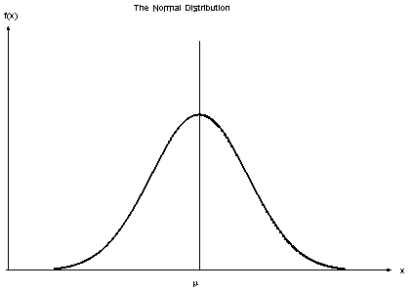
Evidence extraction

```
call:readLine type:FileReader type:BufferedReader
type:Iterator call:next call:remove
```

```
void remove(List<String> list) {
  Iterator<String> i1;
  boolean b1;
  i1 = list.iterator();
  while ((b1 = i1.hasNext())) {
    i1.remove();
  }
  return;
}
```

Training

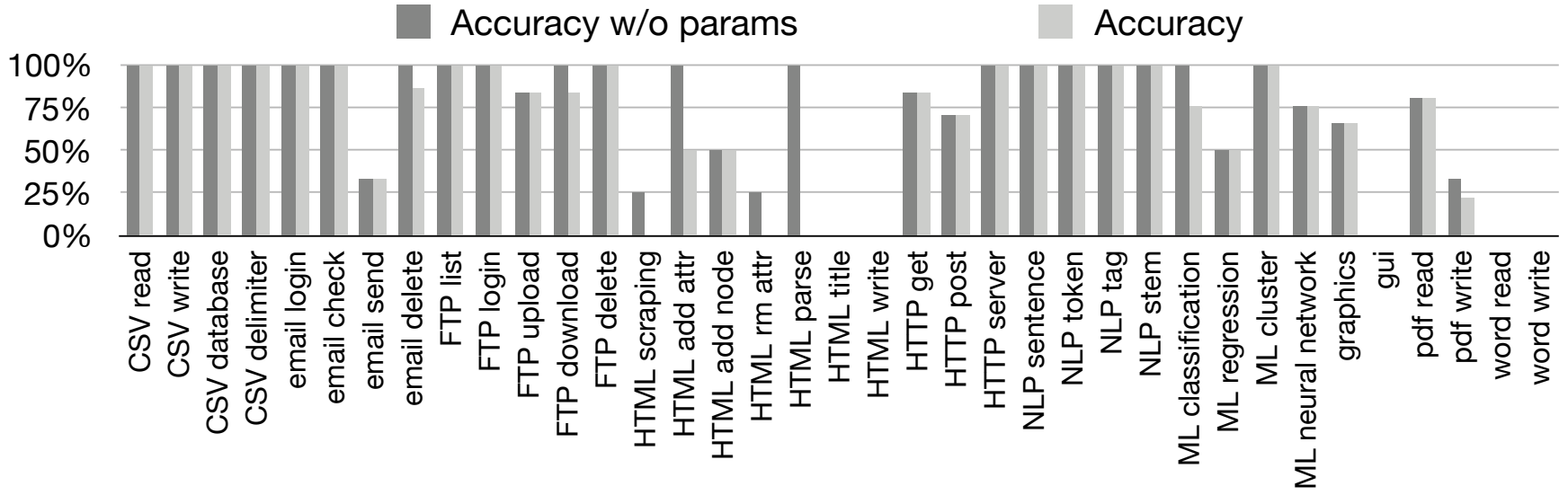
```
call:parse type:Jsoup
call:select call:first
```



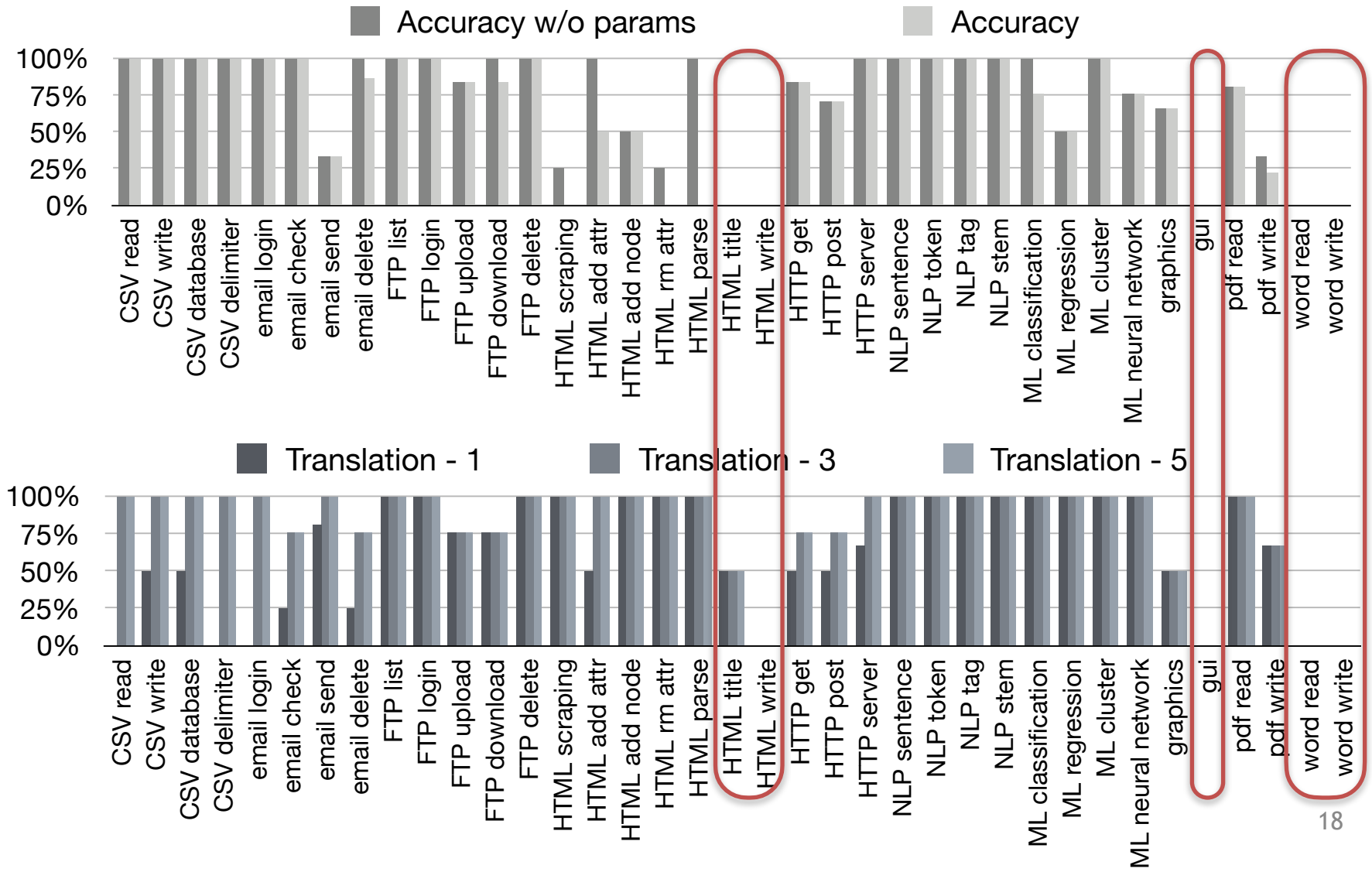

```
void read_href(String content,
               String selector,
               String attr,
               Evaluator _arg01) {
  Elements e1;
  Document d1;
  Elements e3;
  Element e2;
  d1 = Jsoup.parse(content);
  e1 = d1.select(selector);
  e2 = e1.first();
  e3 = Collector.collect(_arg01, e2);
}
```

Evaluation

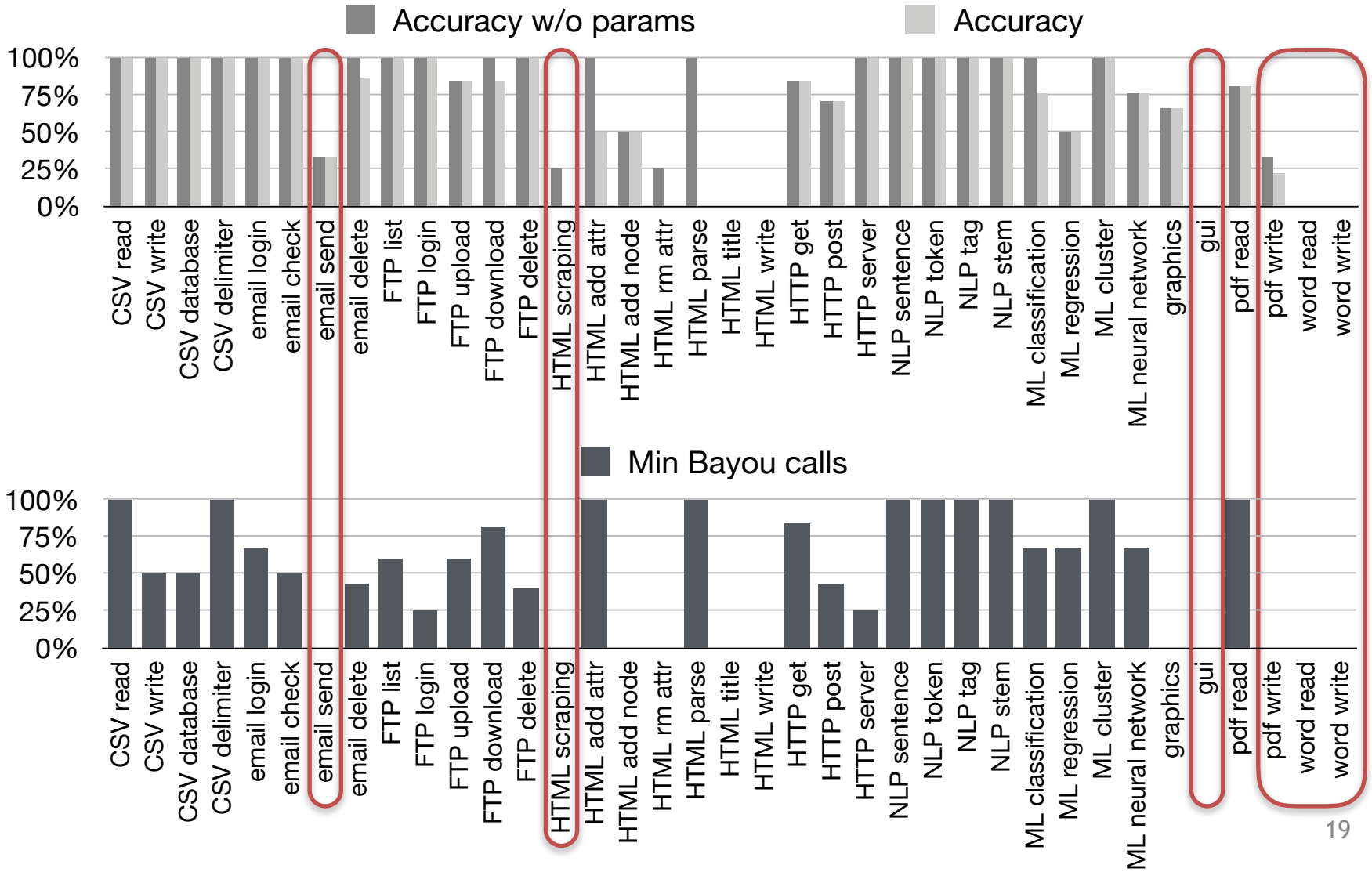
- Accuracy - percentage of correctly generated API calls
- 75% accuracy on most benchmark problems



Translation failure



Rare sequence



API refactoring

- Effective method that automates the process of API refactoring
- Combination of two techniques
  - API call translation
  - API call sequence synthesizer
- Does not work when
  - Terminologies are different
  - Rare sequence