The University of Texas at Austin

Using Developer Discussions to Guide Fixing Bugs in Software

Findings of EMNLP 2022

Sheena Panthaplackel, Milos Gligoric, Junyi Jessy Li, Raymond J. Mooney

Fixing Bugs in Software

Buggy Code

```
sb.append("Invalid table definition due to
            empty implicit table name: ")
  .append(table)
  .append("\n");
```

+

Full Buggy Method

```
void emptyImplicitTable(String table, int line) {
  sb.append("Invalid table definition due to
        empty implicit table name: ")
    .append(table)
    .append("\n");
}
```

+

Natural Language Context

Removed trailing newlines from error messages

Automated Bug-Fixing Model

Fixed Code

```
sb.append("Invalid table definition due to
            empty implicit table name: ")
  .append(table);
```

- Extremely challenging task with such limited context
- MODIT *[Chakraborty and Ray, 2021]* incorporates two additional sources of input

[Le Goues et al., 2012; Kim et al., 2013; Ke et al., 2015; Le et al., 2017; Tufano et al., 2019; Chen et al., 2019; Lutellier et al., 2020; Mashhadi and Hemmati, 2021; Allamanis et al., 2021; Chakraborty and Ray, 2021]

Sources of Natural Language Context

Natural Language Context
Removed trailing newlines from error messages

MODIT *[Chakraborty and Ray, 2021]*:

- Requires prompting developers
  Burdensome for developers
- Simulated through oracle commit messages
  Inaccurately reflect the available context since they are written after the bug-fixing commits

Is there a source of naturally-occurring natural language context that is available before the task is to be performed?

Bug Report Discussions

Title: Parsing exception messages contain trailing newlines

Utterance #1
Some of the parsing exceptions thrown by toml4j contains trailing newlines. This is somewhat unusual, and causes empty lines in log files when the exception messages are logged…

Utterance #2
The idea was to be able to display multiple error messages at once. However, processing stops as soon as an error is encountered, so that's not even possible. Removing the newlines shouldn't be a problem, then.

Deriving NL Context from Bug Report Discussions

Title: Parsing exception messages contain trailing newlines

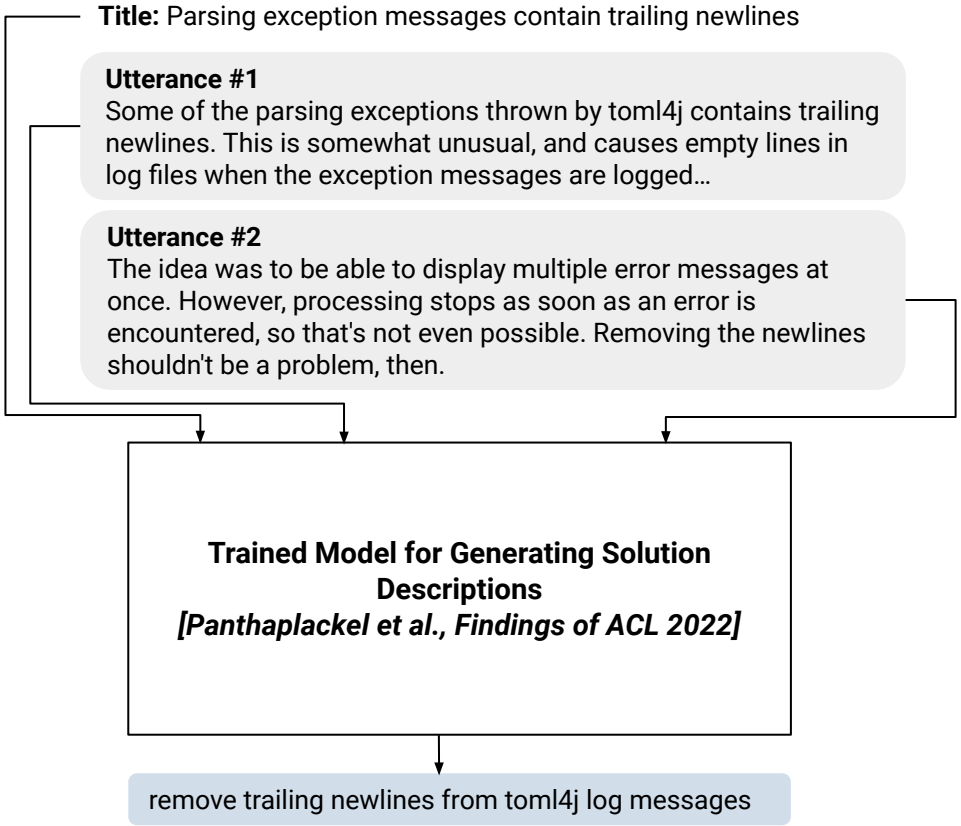
Utterance #1
Some of the parsing exceptions thrown by toml4j contains trailing newlines. This is somewhat unusual, and causes empty lines in log files when the exception messages are logged…

Utterance #2
The idea was to be able to display multiple error messages at once. However, processing stops as soon as an error is encountered, so that's not even possible. Removing the newlines shouldn't be a problem, then.

Deriving Context **Heuristically**

- Whole discussion
- Title
- Last utterance

Deriving NL Context from Bug Report Discussions



Deriving Context **Heuristically**

- Whole discussion
- Title
- Last utterance

Deriving Context **Algorithmically**

- Solution description
- Solution description + title
- Attended segments

Approach

Buggy Code

```
sb.append("Invalid table definition due to
            empty implicit table name: ")
  .append(table)
  .append("\n");
```

+

Full Buggy Method

```
void emptyImplicitTable(String table, int line) {
  sb.append("Invalid table definition due to
        empty implicit table name: ")
    .append(table)
    .append("\n");
}
```

+

Natural Language Context

Removed trailing newlines from error messages

Automated Bug-Fixing Model

Fixed Code

```
sb.append("Invalid table definition due to
            empty implicit table name: ")
  .append(table);
```


MODIT [Chakraborty and Ray, 2021]

Approach

Buggy Code

```
sb.append("Invalid table definition due to
            empty implicit table name: ")
  .append(table)
  .append("\n");
```

+

Full Buggy Method

```
void emptyImplicitTable(String table, int line) {
  sb.append("Invalid table definition due to
        empty implicit table name: ")
    .append(table)
    .append("\n");
}
```

+

Natural Language Context

Removed trailing newlines from error messages

PLBART
[Ahmad et al., 2021]

Fixed Code

```
sb.append("Invalid table definition due to
            empty implicit table name: ")
  .append(table);
```

Approach

Buggy Code

```
sb.append("Invalid table definition due to
            empty implicit table name: ")
  .append(table)
  .append("\n");
```

+

Full Buggy Method

```
void emptyImplicitTable(String table, int line) {
  sb.append("Invalid table definition due to
        empty implicit table name: ")
    .append(table)
    .append("\n");
}
```

+

Natural Language Context

Removed trailing newlines from error messages

Oracle commit message

→ **PLBART** *[Ahmad et al., 2021]* →

Fixed Code

```
sb.append("Invalid table definition due to
            empty implicit table name: ")
  .append(table);
```


MODIT [Chakraborty and Ray, 2021]

Approach

Buggy Code

```
sb.append("Invalid table definition due to
            empty implicit table name: ")
  .append(table)
  .append("\n");
```

+

Full Buggy Method

```
void emptyImplicitTable(String table, int line) {
  sb.append("Invalid table definition due to
        empty implicit table name: ")
    .append(table)
    .append("\n");
}
```

+

Natural Language Context

~~Removed trailing newlines from error messages~~

~~**Oracle commit message**~~

PLBART
[Ahmad et al., 2021]

Fixed Code

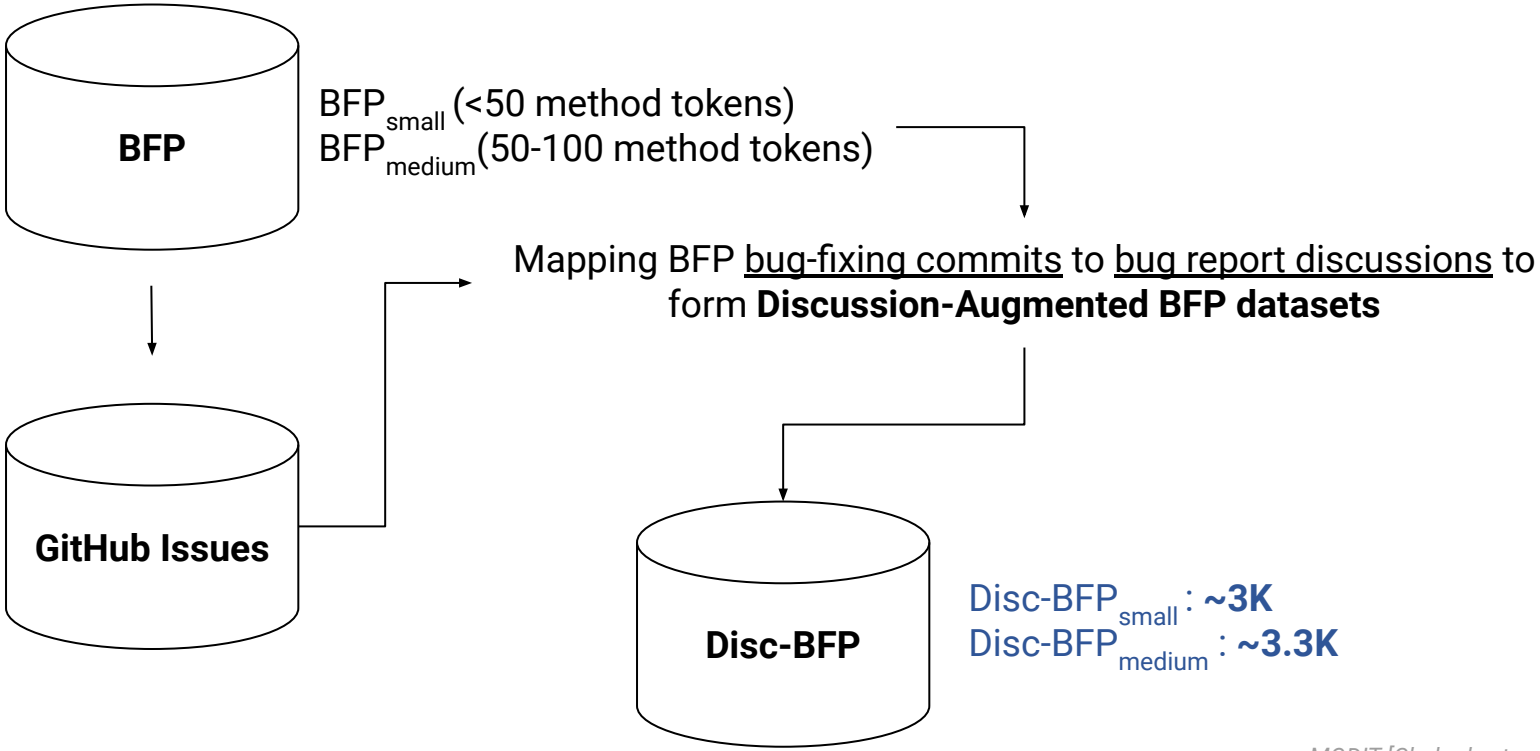
```
sb.append("Invalid table definition due to
            empty implicit table name: ")
  .append(table);
```

NL Context derived from bug report discussion:

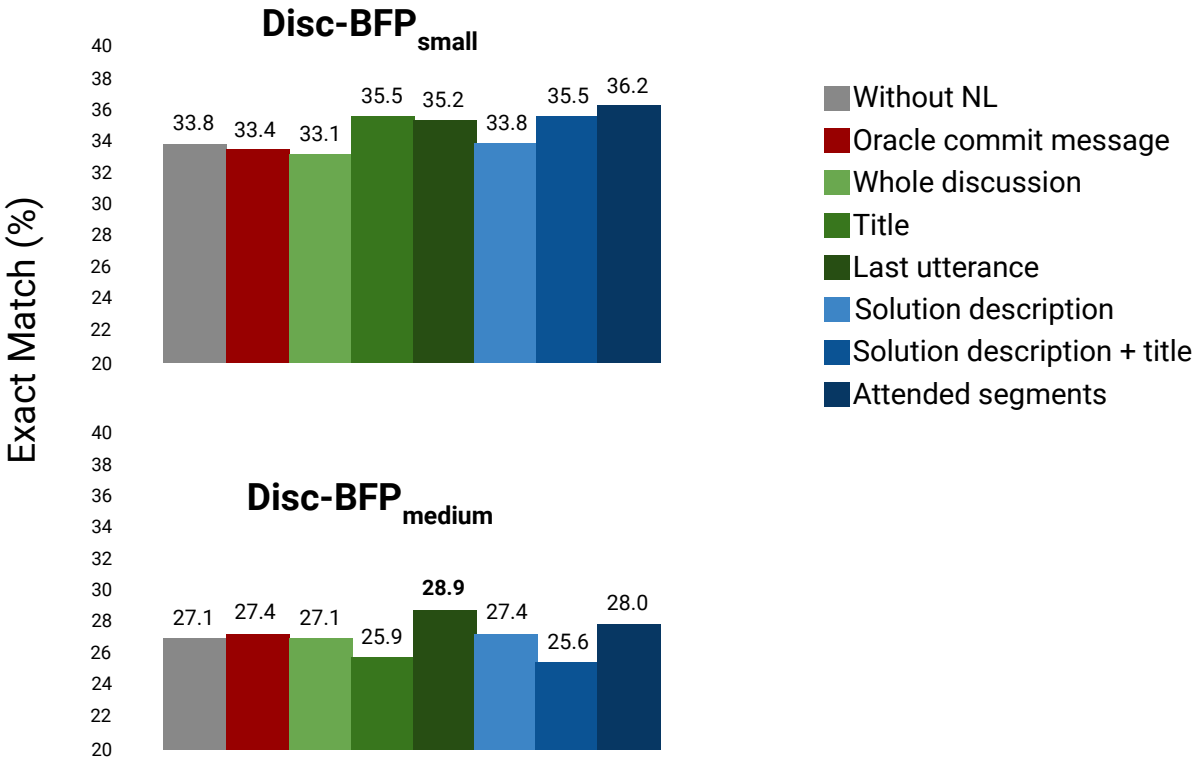
- Whole discussion
- Title
- Last utterance
- Solution description
- Solution description + title
- Attended segments

Data

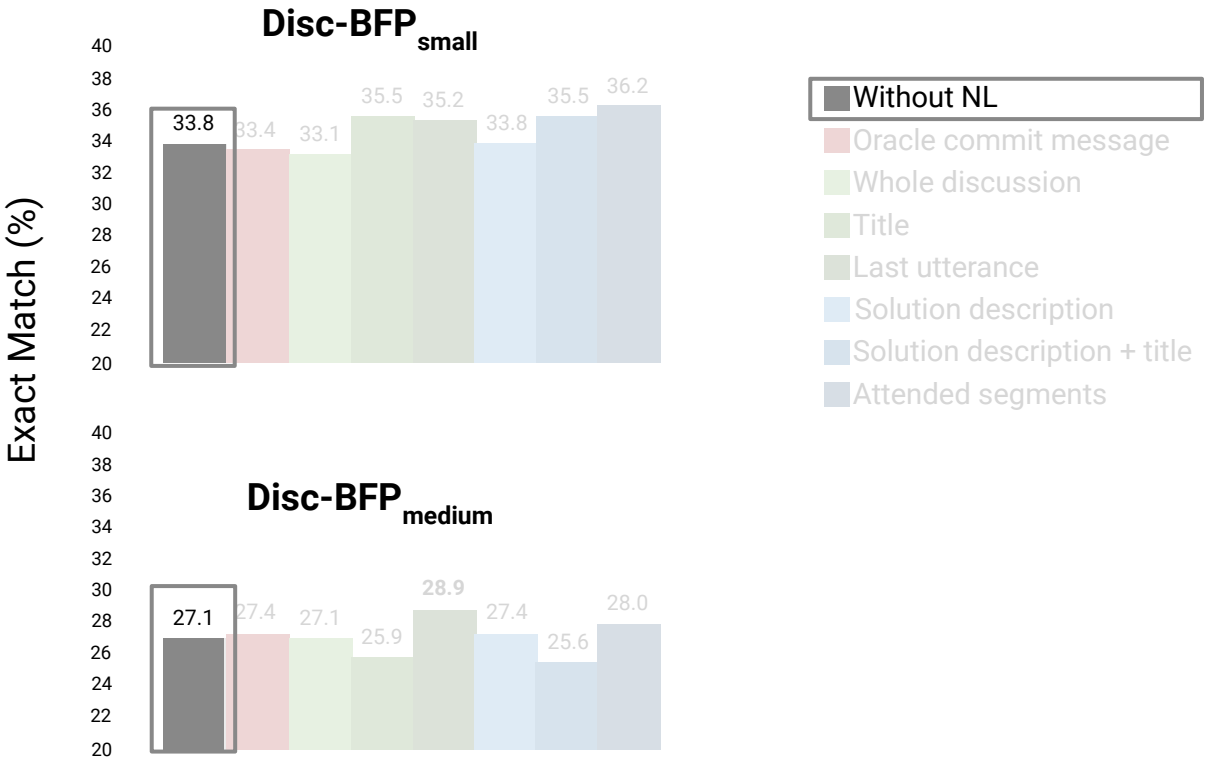
MODIT is built using the **Bug-Fix Patches (BFP) datasets** *[Tufano et al., 2019]*



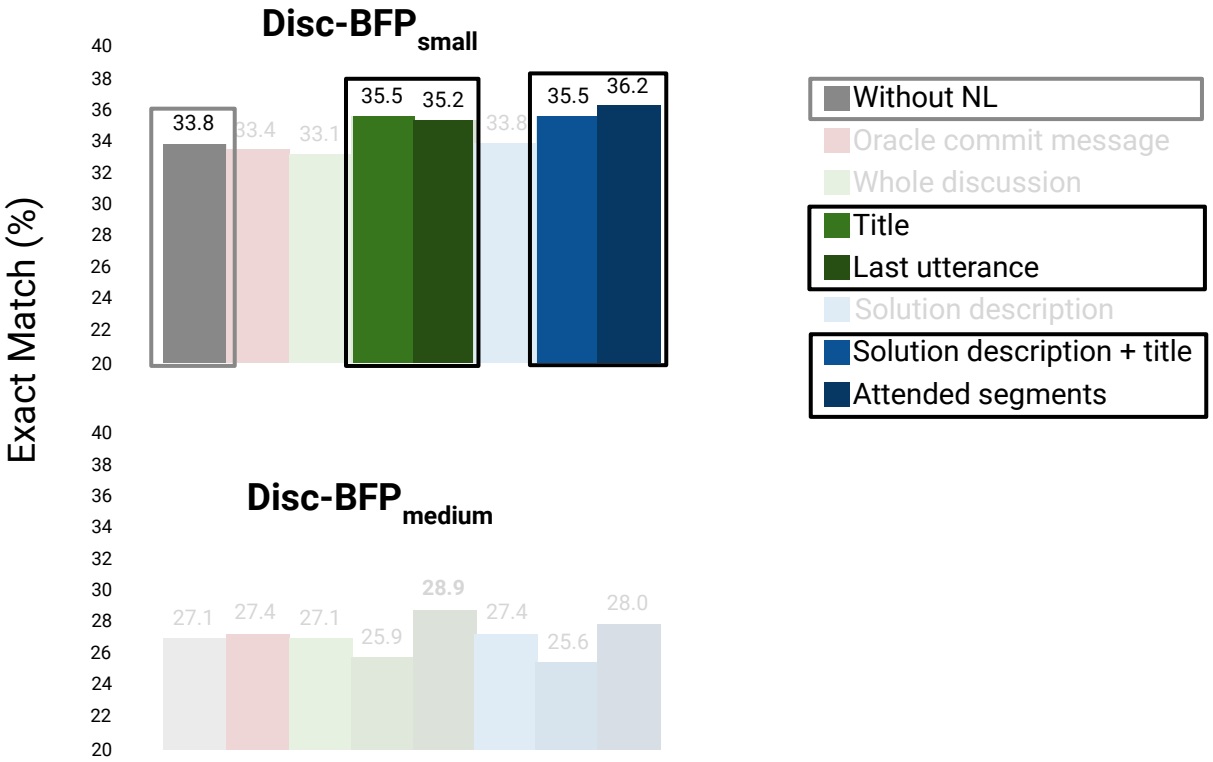
Results


Disc-BFP small
Exact Match (%)
40
38
36
34
32
30
28
26
24
22
20
33.8
33.4
33.1
35.5
35.2
33.8
35.5
36.2
Disc-BFP medium
27.1
27.4
27.1
25.9
28.9
27.4
25.6
28.0
Without NL
Oracle commit message
Whole discussion
Title
Last utterance
Solution description
Solution description + title
Attended segments

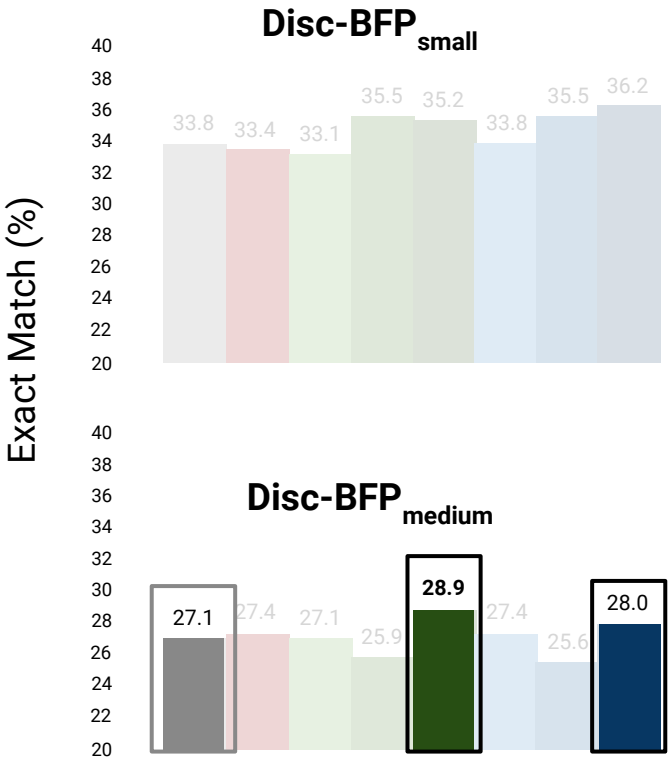
Results


Disc-BFP small
Disc-BFP medium
Exact Match (%)
40
38
36
34
32
30
28
26
24
22
20
33.8
33.4
33.1
35.5
35.2
33.8
35.5
36.2
27.1
27.4
27.1
25.9
28.9
27.4
25.6
28.0
Without NL
Oracle commit message
Whole discussion
Title
Last utterance
Solution description
Solution description + title
Attended segments

Results

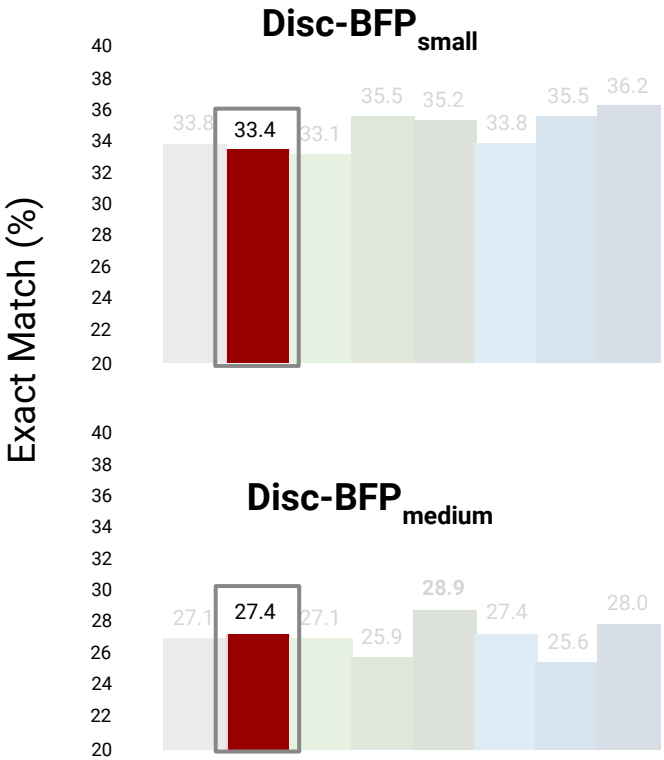

Disc-BFP small
Exact Match (%)
40
38
36
34
32
30
28
26
24
22
20
33.8
33.4
33.1
35.5
35.2
33.8
35.5
36.2
Disc-BFP medium
27.1
27.4
27.1
25.9
28.9
27.4
25.6
28.0
Without NL
Oracle commit message
Whole discussion
Title
Last utterance
Solution description
Solution description + title
Attended segments

Results



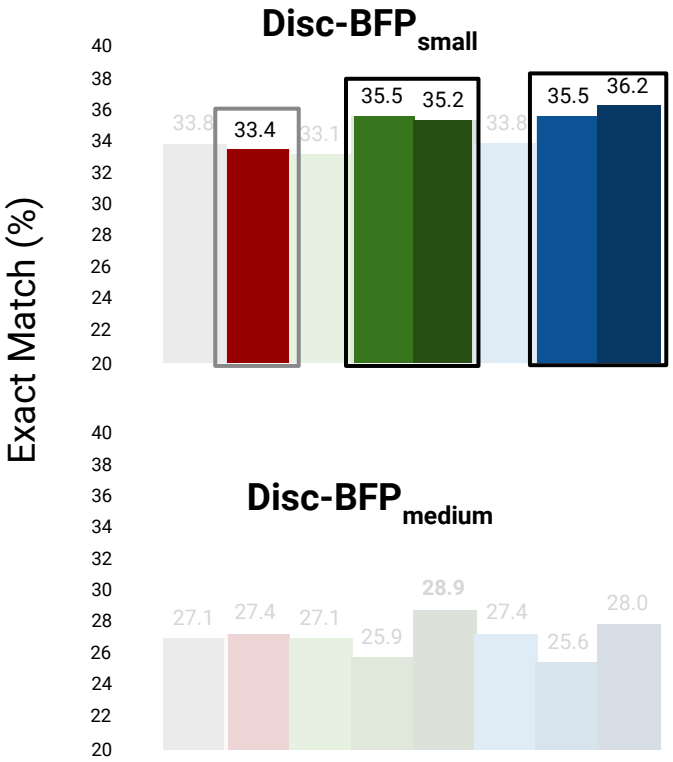
- NL context from bug report discussions yields improvement over baselines without NL

Results



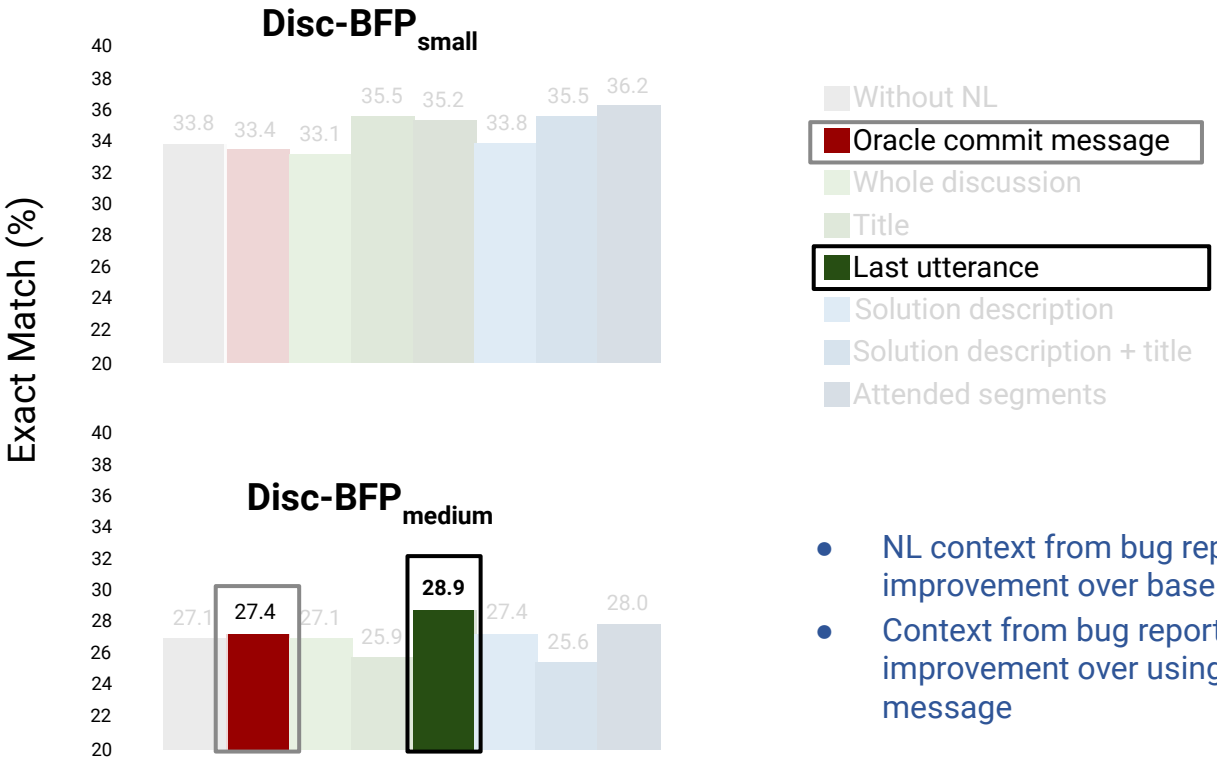
- NL context from bug report discussions yields improvement over baselines without NL

Results



- NL context from bug report discussions yields improvement over baselines without NL

Results



- NL context from bug report discussions yields improvement over baselines without NL
- Context from bug report discussions yields improvement over using the oracle commit message

Summary

- Investigated the utility of bug report discussions for automated bug fixing
- Explored various strategies for deriving natural language context from bug report discussions
- Compiled discussion-augmented bug-fixing datasets
- Demonstrated the benefits of using context from bug report discussions, even showing an advantage over using oracle commit messages

Contact: spantha@utexas.edu
Data: https://github.com/panthap2/developer-discussions-for-bug-fixing