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

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

Full Buggy Method

```java
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

Fixed Code

```java
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
  
- 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.
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

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.

Trained Model for Generating Solution Descriptions
[Panthaplackel et al., Findings of ACL 2022]

Deriving Context **Heuristically**
- Whole discussion
- Title
- Last utterance

Deriving Context **Algorithmically**
- Solution description
- Solution description + title
- Attended segments
Approach

Buggy Code

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

Fixed Code

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

Full Buggy Method

```java
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

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

Fixed Code

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

PLBART [Ahmad et al., 2021]

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

Fixed Code

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

PLBART [Ahmad et al., 2021]
Approach

Buggy Code

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

Fixed Code

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

Full Buggy Method

```java
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

MODIT [Chakraborty and Ray, 2021]
Approach

Buggy Code

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

Fixed Code

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

Full Buggy Method

```java
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]

Oracle commit message

NL Context derived from bug report discussion:
- Whole discussion
- Title
- Last utterance
- Solution description
- Solution description + title
- Attended segments
MODIT is built using the **Bug-Fix Patches (BFP) datasets** [Tufano et al., 2019]

- **BFP**
  - **BFP\textsubscript{small}** (<50 method tokens)
  - **BFP\textsubscript{medium}** (50-100 method tokens)

Mapping BFP bug-fixing commits to bug report discussions to form **Discussion-Augmented BFP datasets**

- **Disc-BFP\textsubscript{small}**: ~3K
- **Disc-BFP\textsubscript{medium}**: ~3.3K
Results

<table>
<thead>
<tr>
<th></th>
<th>Without NL</th>
<th>Oracle commit message</th>
<th>Whole discussion</th>
<th>Title</th>
<th>Last utterance</th>
<th>Solution description</th>
<th>Solution description + title</th>
<th>Attended segments</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Disc-BFP small</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Exact Match (%)</td>
<td>33.8</td>
<td>33.4</td>
<td>33.1</td>
<td>35.5</td>
<td>35.2</td>
<td>33.8</td>
<td>35.5</td>
<td>36.2</td>
</tr>
<tr>
<td><strong>Disc-BFP medium</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Exact Match (%)</td>
<td>27.1</td>
<td>27.4</td>
<td>27.1</td>
<td>25.9</td>
<td>28.9</td>
<td>27.4</td>
<td>25.6</td>
<td>28.0</td>
</tr>
</tbody>
</table>
Results

**Disc-BFP**

**small**

- 33.8
- 33.4
- 35.5
- 35.2
- 33.8
- 35.5
- 36.2

**medium**

- 27.1
- 27.1
- 27.4
- 25.9
- 28.9
- 27.4
- 25.6
- 28.0

- 20
- 22
- 24
- 26
- 28
- 30
- 32
- 34
- 36
- 38
- 40

**Without NL**

- Oracle commit message
- Whole discussion
- Title
- Last utterance
- Solution description
- Solution description + title
- Attended segments
Results

Disc-BFP\textsubscript{small}:
- Exact Match (%): 33.8, 33.4, 33.1, 35.5, 35.2, 13.8, 35.5, 36.2

Disc-BFP\textsubscript{medium}:
- Exact Match (%): 27.1, 27.4, 27.1, 25.9, 28.9, 27.4, 25.6, 28.0

Legend:
- Without NL
- Oracle commit message
- Whole discussion
- Title
- Last utterance
- Solution description
- Solution description + title
- Attended segments
Results

- **Disc-BFP**
  - small: 33.8, 33.4, 33.1, 35.5, 35.2, 33.8, 35.5, 36.2
  - medium: 27.1, 27.4, 27.1, 25.9, 27.4, 25.6, 28.9, 28.0

- **Oracle commit message**
- **Whole discussion**
- **Title**

- **Last utterance**
- **Solution description**
- **Solution description + title**
- **Attended segments**

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

- **Disc-BFP**
  - **small**
    - Without NL:
      - Whole discussion: 33.1%
      - Title: 35.2%
      - Last utterance: 33.8%
      - Solution description: 35.5%
      - Solution description + title: 36.2%
  - **medium**
    - Without NL:
      - Whole discussion: 27.1%
      - Title: 28.9%
      - Last utterance: 27.4%
      - Solution description: 25.6%
      - Solution description + title: 28.0%

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

<table>
<thead>
<tr>
<th></th>
<th>Disc-BFP <em>small</em></th>
<th>Disc-BFP <em>medium</em></th>
</tr>
</thead>
<tbody>
<tr>
<td>Exact Match (%)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Oracle commit message</td>
<td>33.4</td>
<td>33.8</td>
</tr>
<tr>
<td>Whole discussion</td>
<td>35.5</td>
<td>35.5</td>
</tr>
<tr>
<td>Title</td>
<td>33.1</td>
<td>27.1</td>
</tr>
<tr>
<td>Last utterance</td>
<td>35.2</td>
<td>27.4</td>
</tr>
<tr>
<td>Solution description</td>
<td>33.8</td>
<td>25.6</td>
</tr>
<tr>
<td>Solution description + title</td>
<td>35.5</td>
<td>25.6</td>
</tr>
<tr>
<td>Attended segments</td>
<td>36.2</td>
<td>28.0</td>
</tr>
</tbody>
</table>

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