SOTorrent

MSR Mining Challenge 2019

Sebastian Baltes
@s_baltes

empirical-software.engineering

16th International Conference on Mining Software Repositories (MSR 2019)
May 26-27, Montreal, Canada
Thanks to my co-chairs!

Christoph Treude  Stephan Diehl
Thanks to all PC members!
Thanks to all authors!
MSR Mining Challenge 2019: Submissions

• 38 abstracts and 27 papers submitted
• 14 papers accepted (52% acceptance rate)
• Diverse range of topics covered, including
  • Quality of code snippets
  • Analysis of duplicate content
  • Evolution of content on Stack Overflow
MSR Mining Challenge 2019: Changes

• Presentation award → Student presentation award

• New open science policy:
  • Authors are encouraged to share their scripts and data in a replication package
  • Replication packages should be published on a preserved archive such as Figshare or Zenodo to receive a DOI
  • Cite replication package in paper using DOI
  • Please remember that GitHub is not a preserved archive (but repos can be automatically archived on Zenodo)
  • Only papers adhering to the open science policy qualify for the best paper award
MSR Mining Challenge 2019: Impact of Changes

• Mining Challenge 2018 (13 accepted papers)
  • 7 (54%) papers did not share any software or data
  • 4 (31%) papers provided links to non-preserved archives, of which 2 were already dead (as of September 2018)
  • 2 (15%) papers provided replication packages on preserved archives

• Mining Challenge 2019 (14 accepted papers)
  • 5 (36%) papers did not share any software or data
  • 0 (0%) papers provided links to non-preserved archives
  • 9 (64%) papers provided replication packages on preserved archives (8 Zenodo, 1 Figshare)
MSR Mining Challenge 2019: Program

1. SOTorrent: Studying the Origin, Evolution, and Usage of Stack Overflow Code Snippets
2. Mining Rule Violations in JavaScript Code Snippets
4. Man vs Machine - A Study into language identification of Stackoverflow code snippets
5. Python Coding Style Compliance on Stack Overflow
6. Towards Mining Answer Edits to Extract Evolution Patterns in Stack Overflow
7. Analyzing Comment-induced Updates on Stack Overflow
8. What Edits Are Done on Highly Answered Stack Overflow Questions? An Empirical Study
9. Can Duplicate Posts on Stack Overflow Benefit the Software Development Community?
10. How Often and What StackOverflow Posts Do Developers Reference in Their GitHub Projects?
11. Characterizing Duplicate Code Snippets between Stack Overflow and Tutorials
12. Challenges with Responding to Static Analysis Tool Alerts
13. Impact of stack overflow code snippets on software cohesion: a preliminary study
14. We Need to Talk about Microservices: an Analysis from the Discussions on StackOverflow
15. What do developers know about machine learning: a study of ML discussions on StackOverflow

Properties of Snippets

Content Evolution

Clones and References

Other
MSR Mining Challenge 2019: Program

1. **SOTorrent: Studying the Origin, Evolution, and Usage of Stack Overflow Code Snippets**
2. Mining Rule Violations in JavaScript Code Snippets
4. Man vs Machine - A Study into language identification of Stackoverflow code snippets
5. Python Coding Style Compliance on Stack Overflow
6. Towards Mining Answer Edits to Extract Evolution Patterns in Stack Overflow
7. Analyzing Comment-induced Updates on Stack Overflow
8. What Edits Are Done on Highly Answered Stack Overflow Questions? An Empirical Study
9. Can Duplicate Posts on Stack Overflow Benefit the Software Development Community?
10. How Often and What StackOverflow Posts Do Developers Reference in Their GitHub Projects?
11. Characterizing Duplicate Code Snippets between Stack Overflow and Tutorials
12. Challenges with Responding to Static Analysis Tool Alerts
13. Impact of stack overflow code snippets on software cohesion: a preliminary study
14. We Need to Talk about Microservices: an Analysis from the Discussions on StackOverflow
15. What do developers know about machine learning: a study of ML discussions on StackOverflow
SOTorrent

Studying the Origin, Evolution, and Usage of Stack Overflow Code Snippets

Sebastian Baltes

@s_baltes

empirical-software.engineering

16th International Conference on Mining Software Repositories (MSR 2019)
May 26-27, Montreal, Canada
Why Reconstruct and Analyze SO Post Evolution?

• The content of **14.8 million posts** has been **edited** after creation (SOTorrent 2019-03-17)

• Like other **software artifacts**, SO posts **evolve over time**:
  • Bugs in code snippets are fixed
  • Clarifications are added in text documenting the code
  • Snippets are updated to new language/library versions

• **Copying code** from Stack Overflow (SO) is common, despite licensing, security, and maintainability implications
Why Reconstruct and Analyze SO Post Evolution?

Evolution of code on SO differs from regular software projects:

- **Short** code snippets (12 LOC on average)
- **No bug tracking** system (just comments and new answers)
- **No versioning** for individual snippets (just whole posts)
Example

Question

https://stackoverflow.com/q/309424

Answer

https://stackoverflow.com/a/5445161
Here's a way using only standard Java library (note that the stream is not closed, YMMV).

```java
static String convertStreamToString(java.io.InputStream is) {
  java.util.Scanner s = new java.util.Scanner(is).useDelimiter("\A");
  return s.hasNext() ? s.next() : "";
}
```

I learned this trick from "Stupid Scanner tricks" article. The reason it works is because Scanner iterates over tokens in the stream, and in this case we separate tokens using "beginning of the input boundary" (A) thus giving us only one token for the entire contents of the stream.

**Note,** if you need to be specific about the input stream's encoding, you can provide the second argument to Scanner constructor that indicates what charset to use (e.g. "UTF-8").

Hat tip goes also to Jacob, who once pointed me to the said article.

**EDITED:** Thanks to a suggestion from Patrick, made the function more robust when handling an empty input stream. **One more edit:** nixed try/catch, Patrick's way is more laconic.
Here's a way using only standard Java library (note that the stream is not closed, YMMV).

```java
static String convertStreamToString(java.io.InputStream is) {
    java.util.Scanner s = new java.util.Scanner(is).useDelimiter("\A");
    return s.hasNext() ? s.next() : "";
}
```

I learned this trick from "Stupid Scanner tricks" article. The reason it works is because Scanner iterates over tokens in the stream (case we separate tokens using "boundary" (\A) thus giving the entire contents of the stream.

Note, if you need to be specific about the input stream's encoding, you can provide the second argument to Scanner constructor that indicates what charset to use (e.g. "UTF-8").

Hat tip goes also to Jacob, who once pointed me to the said article.

**EDITED:** Thanks to a suggestion from Patrick, made the function more robust when handling an empty input stream. **One more edit:** nixed try/catch, Patrick's way is more laconic.
Comments

This stuff is clearly a hack.
Even for such a simple code snippet, the context is quite complex:

• The snippet is based on an external source

• Hidden in the comments, the author acknowledges that his solution is “clearly a hack”

• There are several bug reports pointing to issues

• Has the snippet been edited to fix those issues?

• Is the snippet safe to use?
SO Revisions

Problems:
• Version history is only available on the level of whole posts, thus individual code snippets hard to trace
• Comments and edits are not linked
• Not visible how external sources are related

https://stackoverflow.com/posts/5445161/revisions
Open dataset based on the official Stack Overflow data dump

**Problem:**

- Version history only available on the level of whole posts
- Analysis of individual code snippets
- Relation to external resources

**SOTorrent:**

- Reconstructed evolution of individual post blocks
- Possible
- Supports researchers in analyzing links from/to posts
Retrieve all versions of a code snippet:

```
SELECT PostHistoryId, Content, Length, LineCount, PredSimilarity
FROM PostBlockVersion
WHERE PostId=5445161 AND LocalId=2 AND PredEqual=0
ORDER BY PostHistoryId DESC;
```
Retrieve line-based difference for latest version:

```
SELECT PostHistoryId, LocalId, PredLocalId, PostBlockDiffOperationId, Text
FROM PostBlockDiff
WHERE PostHistoryId=155295527 AND LocalId=2;
```

![Table of Changed lines]

<table>
<thead>
<tr>
<th>PostHistoryId</th>
<th>LocalId</th>
<th>PredLocalId</th>
<th>PostBlockDiffOperationId</th>
<th>Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>155295527</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>155295527</td>
<td>2</td>
<td>2</td>
<td>-1</td>
<td></td>
</tr>
<tr>
<td>155295527</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>155295527</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

Equal

Delete

Insert

Equal
Retrieve links from a post version:

```
SELECT PostId, PostHistoryId, Domain, Url
FROM PostVersionUrl
WHERE PostHistoryId=155295527;
```
Retrieve links from GitHub repos to post:

```
SELECT PostId, RepoName, Branch, Path, FileExt, Size, Copies
FROM PostReferenceGH
WHERE PostId=5445161;
```

Referenced in 113 distinct repos
SOTorrent: Studying the Origin, Evolution, and Usage of Stack Overflow Code Snippets

Sebastian Baltes  
University of Trier, Germany  
research@sbaltes.com

Christoph Treude  
University of Adelaide, Australia  
christoph.treude@adelaide.edu.au

Stephan Diehl  
University of Trier, Germany  
diehl@uni-trieber.de

Abstract—Stack Overflow (SO) is the most popular question-and-answer website for software developers, providing a large amount of copyable code snippets. Like other software artifacts, code on SO evolves over time, for example when bugs are fixed or APIs are updated to the most recent version. To be able to analyze how code and the surrounding text on SO evolves, we built SOTorrent, an open dataset based on the official SO data dump. SOTorrent provides access to the version history of SO content at the level of whole posts and individual text and code blocks. It connects code snippets from SO posts to other platforms by aggregating URLs from surrounding text blocks and comments, and by collecting references from GitHub files to SO posts. Our vision is that researchers will use SOTorrent to investigate and understand the evolution and maintenance of code on SO and its relation to other platforms such as GitHub.

dataset [16] that enables researchers to analyze the version history of SO posts at the level of individual text and code blocks (see Figure 1 for exemplary posts). The official SO data dump [1] keeps track of different versions of entire posts, but does not contain information about differences between versions at a more fine-grained level. In particular, extracting different versions of the same code snippet from the history of a post is challenging and required us to develop a complex strategy, involving the evaluation of 134 different string similarity metrics [15]. Beside providing access to the version history, our dataset links SO posts to external resources in two ways: (1) by extracting linked URLs from text blocks of SO posts and from post comments and (2) by providing

sotorrent.org

Dataset available on Zenodo and BigQuery
Voting for Best Student Presentation
MSR Mining Challenge 2019: Program

1. SOTorrent: Studying the Origin, Evolution, and Usage of Stack Overflow Code Snippets
2. Mining Rule Violations in JavaScript Code Snippets
4. Man vs Machine - A Study into language identification of Stackoverflow code snippets
5. Python Coding Style Compliance on Stack Overflow
6. Towards Mining Answer Edits to Extract Evolution Patterns in Stack Overflow
7. Analyzing Comment-induced Updates on Stack Overflow
8. What Edits Are Done on Highly Answered Stack Overflow Questions? An Empirical Study
9. Can Duplicate Posts on Stack Overflow Benefit the Software Development Community?
10. How Often and What StackOverflow Posts Do Developers Reference in Their GitHub Projects?
11. Characterizing Duplicate Code Snippets between Stack Overflow and Tutorials
12. Challenges with Responding to Static Analysis Tool Alerts
13. Impact of stack overflow code snippets on software cohesion: a preliminary study
14. We Need to Talk about Microservices: an Analysis from the Discussions on StackOverflow
15. What do developers know about machine learning: a study of ML discussions on StackOverflow

I vote for paper number 1 to receive the best student presentation award.
MSR Mining Challenge 2019: Program

1. SOTorrent: Studying the Origin, Evolution, and Usage of Stack Overflow Code Snippets
2. **Mining Rule Violations in JavaScript Code Snippets**
4. Man vs Machine - A Study into language identification of Stackoverflow code snippets
5. Python Coding Style Compliance on Stack Overflow
6. Towards Mining Answer Edits to Extract Evolution Patterns in Stack Overflow
7. Analyzing Comment-induced Updates on Stack Overflow
8. What Edits Are Done on Highly Answered Stack Overflow Questions? An Empirical Study
9. Can Duplicate Posts on Stack Overflow Benefit the Software Development Community?
10. How Often and What StackOverflow Posts Do Developers Reference in Their GitHub Projects?
11. Characterizing Duplicate Code Snippets between Stack Overflow and Tutorials
12. Challenges with Responding to Static Analysis Tool Alerts
13. Impact of stack overflow code snippets on software cohesion: a preliminary study
14. We Need to Talk about Microservices: an Analysis from the Discussions on StackOverflow
15. What do developers know about machine learning: a study of ML discussions on StackOverflow
Voting for Best Student Presentation
Banquet

• Forum de Montreal Games Centre
  2313 Rue St-Catherine Ouest

• 3rd floor (escalator/elevator)

• From 18:30 on: drinks, arcades, billiards, air hockey, table tennis!

• From 20:00 on: buffet dinner!

• You need your MSR dinner voucher to attend!

• Please email irina@sians.org if you will not attend the dinner!
1. SOTorrent: Studying the Origin, Evolution, and Usage of Stack Overflow Code Snippets
2. Mining Rule Violations in JavaScript Code Snippets (Uriel Campos)
4. Man vs Machine - A Study into language identification of Stackoverflow code snippets
5. Python Coding Style Compliance on Stack Overflow (Wenjie Boon)
6. Towards Mining Answer Edits to Extract Evolution Patterns in Stack Overflow
7. Analyzing Comment-induced Updates on Stack Overflow
8. What Edits Are Done on Highly Answered Stack Overflow Questions? An Empirical Study (Xianhao Jin)
9. Can Duplicate Posts on Stack Overflow Benefit the Software Development Community? (Durham Abric)
10. How Often and What StackOverflow Posts Do Developers Reference in Their GitHub Projects? (Saraj Singh Manes)
11. Characterizing Duplicate Code Snippets between Stack Overflow and Tutorials (Agnieszka Ciborowska)
12. Challenges with Responding to Static Analysis Tool Alerts (Nasif Imtiaz)
13. Impact of stack overflow code snippets on software cohesion: a preliminary study (Mashal Ahmad)
14. We Need to Talk about Microservices: an Analysis from the Discussions on StackOverflow
15. What do developers know about machine learning: a study of ML discussions on StackOverflow (Abdul Ali Bangash)