## Automatic Data Categorization in Issue Tracking Systems

**A Research Preview**

**Thorsten Merten**\(^1\)  Barbara Paech\(^2\)

\(^1\)Bonn-Rhein-Sieg University of Applied Sciences, Department of Computer Science, Germany

thorsten.merten@h-brs.de, simone.buersner@h-brs.de

\(^2\)University of Heidelberg, Software Engineering Group, Germany

paech@informatik.uni-heidelberg.de

---

### Issue Tracking System Data

- Issue Tracking Systems contain valuable Software Engineering information, e.g.
  - requirement and feature descriptions
  - development and refactoring tasks
  - bug reports and bug fixing tasks
  - discussions and dissent
  - ...  
- This information fits in multiple categories
  - implementation ideas
  - stack traces or error messages
  - social interaction
  - ...

### ITS Data Mining

- Most data in an Issue Tracking System (ITS) is stored in user defined text fields
- Data mining approaches try to extract and use this data
- Categorization of ITS data can support multiple SWE tasks, e.g., cf. right figure
  - creating a new sub-issue if necessary
  - marking or changing wrongly entered manual ITS data
  - marking relevant information for a certain task
  - removing unnecessary information

### Our Approach

**Hypothesize**

How can the task be improved by categorization of ITS data?

**Categorize**

Which tasks in SWE can be supported by ITS and ITS data?

**Identify**

How has the task been influenced?

**Measure**

How can required data for the task be identified?

**Combine algorithms**

use existing algorithms

**General idea:** Improvement of semi-supervised learning by combining multiple methods.

### References


---

### Example for Algorithmic Improvements

- To support automatic creation of change logs
- Some prerequisites can be used
  - The separation of technical items and natural language texts already works for developer mailing list classification [1]
  - Semi-supervised learning can be used to separate relevant change log information but precision and recall are generally low
  - Preprocessing is very important in Text Mining [2, pp. 84]
- We can use the specialized algorithm to separate technical items as preprocessing step for general classification [5, 4]

### Example for Task-Based Utilization

- **Identify**
  - Documentation is a SWE task. The creation of a change log is a sub-discipline, which can be supported by ITS data.
- **Hypothesize**
  - We can improve the categorization of task-relevant and irrelevant natural language text passages by separating technical information.
- **Categorize**
  - We implement different algorithms to separate technical and natural language information and combine those (as preprocessing) with different semi-supervised learning algorithms.
- **Evaluate**
  - We apply the above categorization to create change logs and evaluate by a) comparison with manually written change logs and b) surveying developers. We may now go back to the categorize step and improve the method.

---

### Current and Future Work

- Literature Reviews on algorithms and solutions and ITS supported SWE tasks
- Identification of ITS supported SWE tasks
- Identification of data categories to support these tasks
- Use of rule-based and specialized algorithms for preprocessing
- General idea: Improvement of semi-supervised learning by combining multiple methods.
- Experimental validation of algorithms and measuring improvements on ITS tasks