
RoboTA-common-errors

Release 1.1.3

Peter Crowther

Oct 18, 2022

CONTENTS

1 Installation	3
2 RoboTA Config	5
2.1 API Reference	5
2.1.1 RoboTA config	5
2.1.2 Getting started with RoboTA common errors	6
2.1.3 Running the report	6
2.1.4 Using common errors as a library	6
2.1.5 Adding new common errors	6
2.1.6 robot_common_errors	7
Python Module Index	13
Index	15

RoboTA (Robot Teaching Assistant) is a group of Python packages that provide a framework for the assessment of software engineering practices. The focus of RoboTA is the assessment of student software engineering coursework, though it has a wider scope in the assessment of general good practice in software engineering.

The robota-core package collects information about a project from a number of sources, git repositories, issue trackers, ci-servers. It is designed to be provider agnostic, for example repository data can come from GitLab or GitHub.

The robota-common-errors package then uses this information to assess the project against a number of common software engineering errors or bad practices. Examples include committing non-project files to a git repository or merging a git branch into the remote tracking branch instead of the local one. The included errors are designed to be general and project agnostic but it would be easy to add new methods to enforce project or group specific standards such as a standard format for git commit messages or ensuring that issues are always assigned to an individual as soon as they are opened.

RoboTA was developed in the [Computer Science](#) department at the [University of Manchester](#). From 2018 to March 2021, development of RoboTA was funded by the [Institute of Coding](#).

**CHAPTER
ONE**

INSTALLATION

To install as a Python module, type

```
python -m pip install .
```

from the root directory. For developers, you should install in linked .egg mode using

```
python -m pip install roboata-core -e
```

If you are using a Python virtual environment, you should activate this first before using the above commands.

ROBOTA CONFIG

RoboTA requires access to a number of data sources to collect data to operate on.

Details of these data sources and information required to connect to them is provided in the robota config yaml file.

Documentation on the config file can be found in the *RoboTA config* section of the documentation.

An example robota config file can be found in the [root of the robota-common-errors repository](#).

2.1 API Reference

2.1.1 RoboTA config

RoboTA reads in various types of data from different sources. These are specified in the robota config file. The main data source documentation is included in the [robota-core documentation](#). This documentation is supplementary to that, so you should read that documentation first.

robota-common-errors only makes use of a subset of the possible data types in robota-core and adds a single additional data type.

Data types

Issues, ci, repository and remote_provider are data types from robota-core that are used in robota-common-errors. They are documented in the [robota-core documentation](#).

There is one additional data type for robota-common-errors:

common-errors

The location of the yaml file that defines the common errors.

Valid sources:

- local_path
- gitlab

required keys:

- file_name - The path of the file containing the mapping between names and email addresses

This keys may specify sub-folder(s) in the git repository, e.g.

file_name: config_files/error_definitions.yaml

Data Sources

Data sources are unchanged from roboata-core.

2.1.2 Getting started with RoboTA common errors

Once the roboata-common-errors package has been installed you can either run the provided report which will output detected common errors to a HTML page, or you can import the functions and incorporate the outputs into your own workflow.

2.1.3 Running the report

The common-errors report is not intended as the primary method for displaying the detected common errors, but is provided as a simple example of a possible output.

To run the common errors report, from the common-errors root directory run:

```
python roboata_common_errors/report.py
```

This will produce a report in the `webpages` folder called `common-error-report.html`.

Optional parameters for the script are:

- `--config_path` - This is the path to the roboata config file. The default is `./roboata-config.yaml`
- `--output` - This is the path where the report will be output. The default is `./webpages`
- `--start` - This is the start date of the report. Artifacts (commits, issues etc.) with a creation date older than this will not be considered in the report. The default is **2020-01-01**
- `--end` - This is the end date of the report, Artifacts newer than this will not be considered in the report. The default is **the current date**

2.1.4 Using common errors as a library

If you want to analyse common errors and use the data in your own scripts, you should use the `roboata_common_errors.report.identify_common_errors()` function. This function takes the path to the roboata config, start and end times as arguments and outputs a list of `CommonError`. These can then be parsed into whatever display format you want to use.

2.1.5 Adding new common errors

Common errors are described in the `error_definitions.yaml` file. The `marking_function` key in the error description is the name of the method in the code that will be called to assess the error. The code that actually assesses the error is located in the `common_error_functions.py` file. Functions to assess a common error must have the signature `marking_function_name(data_source: DataServer, common_error: CommonError)` and should populate the `CommonError` with the result and return this.

2.1.6 robota_common_errors

robota_common_errors package

Subpackages

robota_common_errors.output_templates package

Submodules

robota_common_errors.output_templates.build_webpages module

```
robota_common_errors.output_templates.build_webpages(template_dir: Path,  
                                                    output_dir: Path)
```

Create ancillary pages from templates and copy them to *output_dir*.
:param template_dir: The path to load page templates from.
:param output_dir: The path to output generated pages to.

Module contents

Submodules

robota_common_errors.analysis module

An example of a simple data analysis script that might be run on the JSON output of common_error_collection.py.

```
robota_common_errors.analysis.main(file_name: str)  
robota_common_errors.analysis.read_json(file_path: Path) → DataFrame
```

robota_common_errors.common_error_collection module

```
class robota_common_errors.common_error_collection.CommonErrorReport(course: str, year: str,  
                                                                    team_name: str,  
                                                                    error_name: str,  
                                                                    detail_titles: List[str],  
                                                                    error_details:  
                                                                    List[List[str]],  
                                                                    num_occurrences: int)
```

Bases: object

```
course: str  
detail_titles: List[str]  
error_details: List[List[str]]  
error_name: str  
num_occurrences: int
```

```
team_name: str
year: str

robota_common_errors.common_error_collection.main(config_path: str, output_name: str)
robota_common_errors.common_error_collection.output_to_json(file_name: str,
                                                            common_error_reports:
                                                            List[CommonErrorReport])
```

robota_common_errors.common_error_functions module

These functions specifically test the common errors.

```
robota_common_errors.common_error_functions.commit_unnecessary_files(data_source: DataServer,
                                                                    common_error:
                                                                    CommonError) →
                                                                    CommonError
```

Certain files shouldn't be committed to a Git repo, and should probably be specified in a .gitignore file

```
robota_common_errors.common_error_functions.committing_comments(data_source: DataServer,
                                                                common_error: CommonError) →
                                                                CommonError
```

Students often commit commented out code which is bad. This function checks the diffs for any java style commenting syntax.

```
robota_common_errors.common_error_functions.committing_conflict_markup(data_source:
                                                                       DataServer,
                                                                       common_error:
                                                                       CommonError) →
                                                                       CommonError
```

Conflicts should be resolved before committing. Any commit conflict markup which gets committed shows that the conflict has not been properly resolved.

```
robota_common_errors.common_error_functions.confusing_branch_names(data_source: DataServer,
                                                                    common_error:
                                                                    CommonError) →
                                                                    CommonError
```

Some branch names are obviously confusing and should be avoided. e.g. a local branch called origin/branch would have a remote tracking branch called origin/origin/branch

```
robota_common_errors.common_error_functions.get_comment_line_number(diff, line_num_in_diff)

Return starting line number of hunk which contains line_num_in_diff
```

```
robota_common_errors.common_error_functions.get_commit_diffs(commits: List[Commit],
                                                             data_source: DataServer) →
                                                             Dict[Commit, List[Diff]]
```

Get the diffs for each commit in commits.

While it would be fewer API calls to do a diff of all commits at once, the API refuses to return the diff if it is too big. This means doing it one commit at a time is safer.

```
robota_common_errors.common_error_functions.get_line_num_in_diff(diff_up_to_match: Diff, match:
                                                               Match)

Return line number in diff of a regex match object
```

```
robot_a_common_errors.common_error_functions.get_subject_errors(message: str) → List[str]
```

Detect errors with the subject line of git commit messages.

Parameters

message – Commit message

Returns

List of detected problems.

```
robot_a_common_errors.common_error_functions.good_commit_messages(data_source: DataServer,  
                                                               common_error:  
                                                               CommonError) →
```

```
CommonError
```

Git commit messages should have a standardised formatting.

```
robot_a_common_errors.common_error_functions.merge_remote_branch(data_source: DataServer,  
                                                               common_error: CommonError)  
                                                               → CommonError
```

Identify cases where students have merged a remote tracking branch. They should be pulling *before* making local commits or rebasing their local branch onto the remote branch.

```
robot_a_common_errors.common_error_functions.repeated_revert(data_source: DataServer,  
                                                               common_error: CommonError) →  
                                                               CommonError
```

Identify cases where students have repeatedly attempted to revert commits. This is often because they think that a commit revert is like an undo command.

```
robot_a_common_errors.common_error_functions.wrong_use_of_estimate(data_source: DataServer,  
                                                               common_error:  
                                                               CommonError) →  
                                                               CommonError
```

Identify cases where students have misused the */estimate* command when trying to record time estimates for a gitlab issue.

```
robot_a_common_errors.common_error_functions.wrong_use_of_spend(data_source: DataServer,  
                                                               common_error: CommonError)  
                                                               → CommonError
```

Identify cases where students have misused the */spend* command when trying to record time spent working on a gitlab issue.

```
robot_a_common_errors.common_error_functions.wrong_way_merge(data_source: DataServer,  
                                                               common_error: CommonError) →  
                                                               CommonError
```

Identify cases where students have merged develop into their feature branch rather than the feature branch into the development branch.

robota_common_errors.common_errors module

Common errors are suboptimal student behaviors that occur regularly. If a common error is identified, feedback is given on how to correct it or avoid it in future.

Common errors are much like Tasks except they are not assigned any credit.

```
class robot_common_errors.common_errors.CommonError(common_error: dict)
```

Bases: object

Representation of a common error that could be identified

Variables

- **name** – The name of the common error.
- **marking_function** – The name of the function used to assess the common error.
- **description** – A textual description of the common error.
- **detail_titles** – Titles of a table to be printed on the marking report.
- **error_details** – Rows of a table to be printed on the marking report.

```
add_feedback(feedback: List[List[str]])
```

```
add_feedback_titles(feedback_titles: List[str])
```

```
count_errors() → int
```

Sum the occurrences of this error by checking error details.

```
robot_common_errors.common_errors.assess_common_errors(data_source: DataServer, common_errors: List[CommonError]) → List[CommonError]
```

Go through all of the possible common errors, running the assessment function for each one.

```
robot_common_errors.common_errors.get_error_descriptions(robot_config: dict) → List[CommonError]
```

Get the textual description of common errors to identify.

```
robot_common_errors.common_errors.validate_data_sources(data_source: DataServer, common_error: CommonError)
```

Check the data sources that are available from the DataServer.

robota_common_errors.report module

Generates a HTML report of common errors for a git repo.

```
robot_common_errors.report.count_errors(common_errors: List[CommonError]) → int
```

Count how many different common error types were detected.

```
robot_common_errors.report.identify_common_errors(robot_config: dict, start: datetime, end: datetime) → List[CommonError]
```

The main function which gets the common errors.

```
robot_common_errors.report.output_html_report(common_errors: List[CommonError], data_source_info: dict, common_error_summary: dict, output_dir: str)
```

```
robot_a_common_errors.report.run_html_error_report(start: str, end: str, config_path: str, output_dir: str,  
substitution_variables: dict)
```

Get common errors and output them in the form of a HTML report.

```
robot_a_common_errors.report.summarise_common_errors(common_errors: List[CommonError])
```

```
robot_a_common_errors.report.summarise_data_sources(robot_a_config: dict, start: datetime, end:  
datetime) → dict
```

```
robot_a_common_errors.report.update_template(common_errors: List[CommonError], data_source_info:  
dict, common_error_summary: dict)
```

Produce the HTML report by writing the marking results to a HTML template.

Parameters

- **common_errors** – A list of common errors and feedback on them.
- **data_source_info** – A dictionary of information about the data sources that were used.
- **common_error_summary** – A dictionary of stats about the common errors that is printed in the report.

Module contents

- genindex

PYTHON MODULE INDEX

r

 robota_common_errors, 11
 robota_common_errors.analysis, 7
 robota_common_errors.common_error_collection,
 7
 robota_common_errors.common_error_functions,
 8
 robota_common_errors.common_errors, 10
 robota_common_errors.output_templates, 7
 robota_common_errors.output_templates.build_webpages,
 7
 robota_common_errors.report, 10

INDEX

A

add_feedback()

(*robota_common_errors.common_errors.CommonError method*), 10

add_feedback_titles()

(*robota_common_errors.common_errors.CommonError method*), 10

assess_common_errors()

(*in module roboota_common_errors.common_errors*), 10

D

detail_titles(*roboota_common_errors.common_error_collection.CommonError attribute*), 7

E

error_details(*roboota_common_errors.common_error_collection.CommonError attribute*), 7

error_name(*roboota_common_errors.common_error_collection.CommonError attribute*), 7

B

build_webpages()

(*in module roboota_common_errors.output_templates.build_webpages*), 7

C

commit_unnecessary_files()

(*in module roboota_common_errors.common_error_functions*), 8

committing_comments()

(*in module roboota_common_errors.common_error_functions*), 8

committing_conflict_markup()

(*in module roboota_common_errors.common_error_functions*), 8

CommonError

(*class in roboota_common_errors.common_errors*), 10

CommonErrorReport

(*class in roboota_common_errors.common_error_collection*), 7

confusing_branch_names()

(*in module roboota_common_errors.common_error_functions*), 8

count_errors()

(*in module roboota_common_errors.report*), 10

count_errors()

(*robota_common_errors.common_errors.CommonError method*), 10

course(*roboota_common_errors.common_error_collection.CommonErrorReport attribute*), 7

G

get_comment_line_number()

(*in module roboota_common_errors.common_error_functions*), 8

get_commit_diffs()

(*in module roboota_common_errors.common_error_functions*), 8

get_error_descriptions()

(*in module roboota_common_errors.common_errors*), 10

get_line_num_in_diff()

(*in module roboota_common_errors.common_error_functions*), 8

get_subject_errors()

(*in module roboota_common_errors.common_error_functions*), 8

good_commit_messages()

(*in module roboota_common_errors.common_error_functions*), 9

|

identify_common_errors()

(*in module roboota_common_errors.report*), 10

M

main()

(*in module roboota_common_errors.analysis*), 7

main()

(*in module roboota_common_errors.common_error_collection*), 8

merge_error_branch()

(*in module roboota_common_errors.common_error_functions*), 9

module

T robota_common_errors, 11 robota_common_errors.analysis, 7 robota_common_errors.common_error_collection, 7 robota_common_errors.common_error_functions, 8 robota_common_errors.common_errors, 10 robota_common_errors.output_templates, 7 robota_common_errors.output_templates.build_webpages, 7 robota_common_errors.report, 10	J team_name(robota_common_errors.common_error_collection.CommonErrorReport, attribute), 7 update_template() (in module roboota_common_errors.report), 11 V validate_data_sources() (in module roboota_common_errors.common_errors), 10
N	
num_occurrences (roboota_common_errors.common_error_collection.CommonErrorReport, attribute), 7	W wrong_use_of_estimate() (in module roboota_common_errors.common_error_functions), 9
O	
output_html_report() (in module roboota_common_errors.report), 10 output_to_json() (in module roboota_common_errors.common_error_collection), 8	E wrong_use_of_spend() (in module roboota_common_errors.common_error_functions), 9 M wrong_way_merge() (in module roboota_common_errors.common_error_functions), 9
R	
read_json() (in module roboota_common_errors.analysis), 7 repeated_revert() (in module roboota_common_errors.common_error_functions), 9	Y year(robota_common_errors.common_error_collection.CommonErrorReport, attribute), 8
robota_common_errors module, 11 robota_common_errors.analysis module, 7 robota_common_errors.common_error_collection module, 7 robota_common_errors.common_error_functions module, 8 robota_common_errors.common_errors module, 10 robota_common_errors.output_templates module, 7 robota_common_errors.output_templates.build_webpages module, 7 robota_common_errors.report module, 10 run_html_error_report() (in module roboota_common_errors.report), 10	
S	
summarise_common_errors() (in module roboota_common_errors.report), 11 summarise_data_sources() (in module roboota_common_errors.report), 11	