Freie Universität Berlin

Verteidigung der Bachelorarbeit

AG Software Engineering

Entwicklung eines Tools zur Suche nach relevanten Diskussionen aus Github Issue-Trackern für die qualitative Forschung

Robert Selack
4774775
robfu@zedat.fu-berlin.de

Erstgutachter: Prof. Dr. Lutz Prechelt
Zweitgutachter: Prof. Dr. Claudia Müller-Birn
Betreuer: Victor Brekenfeld

Gliederung

1. Einführung
2. Zielstellung der Arbeit
3. Ansätze und Methoden
4. Umgesetztes Tool
5. Fazit und Ausblick

1. Einführung

Technische Schulden:

A → zeitaufwendiger
→ langfristig wenige Modifikationen

B → schnell umsetzbar
→ langfristig viele Modifikationen

→ Technische Schulden

1. Einführung

- Diskussionen → Verhaltensmuster

- Qualitative Forschung

2. Zielstellung der Arbeit

- Entwicklung eines Tools zum Finden solcher Diskussionen

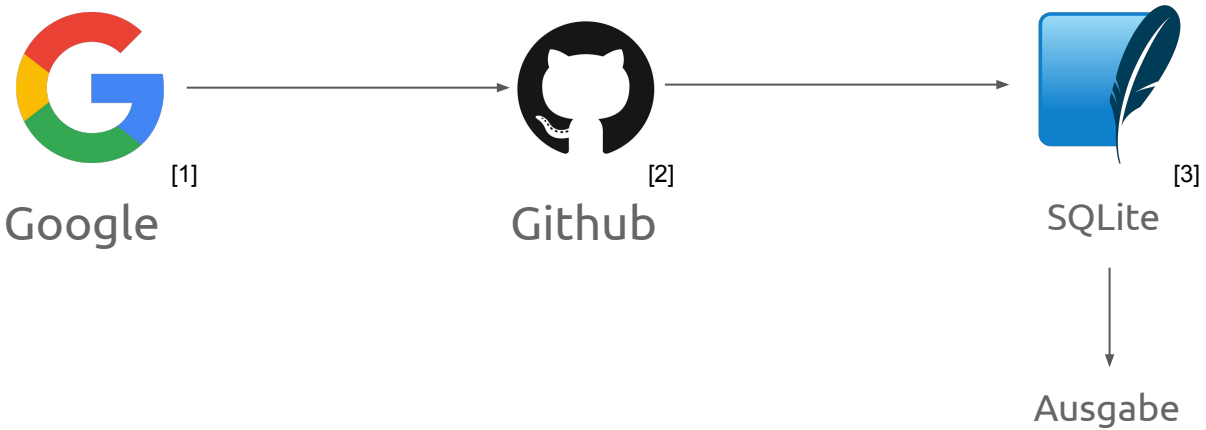
- Benutzerfreundlichkeit

3. Ansätze und Methoden

3.1 Ansatz

3. Ansätze und Methoden

3.1 Ansatz



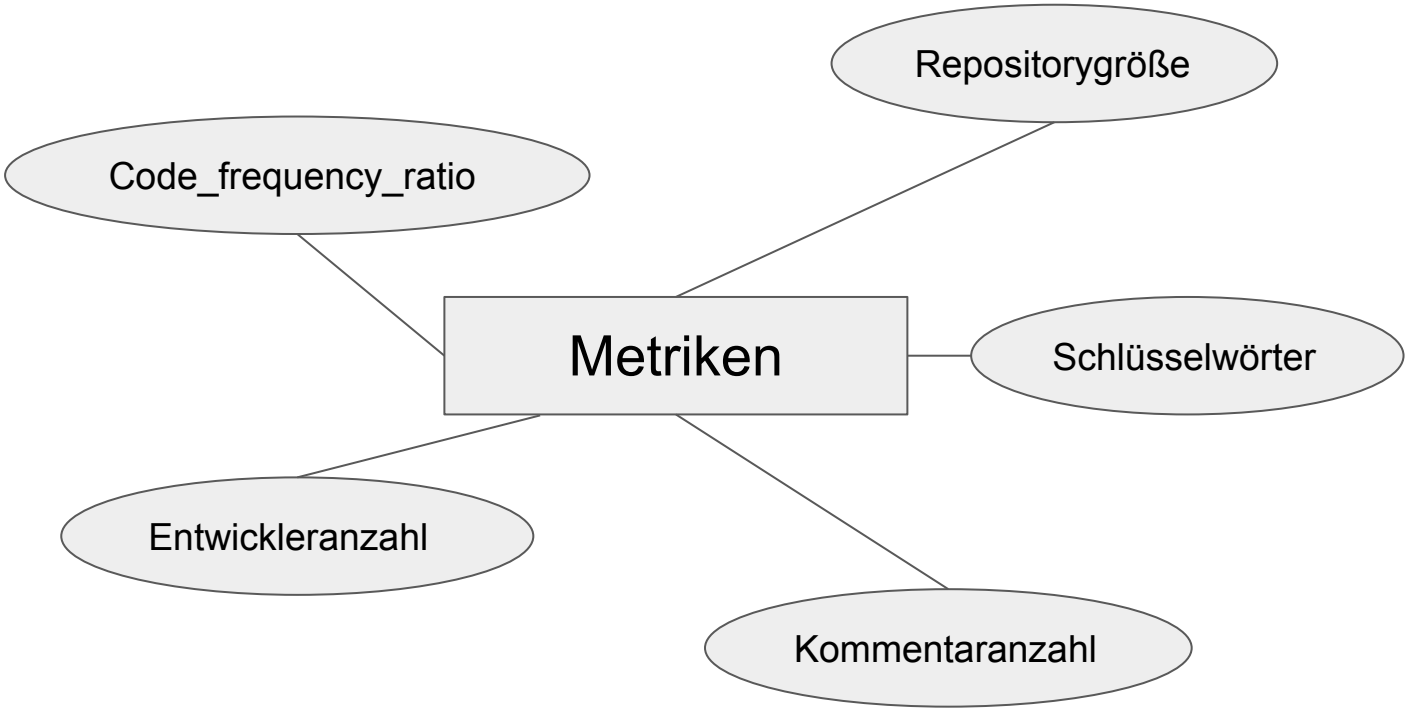
3. Ansätze und Methoden

3.2 Methoden

Metriken

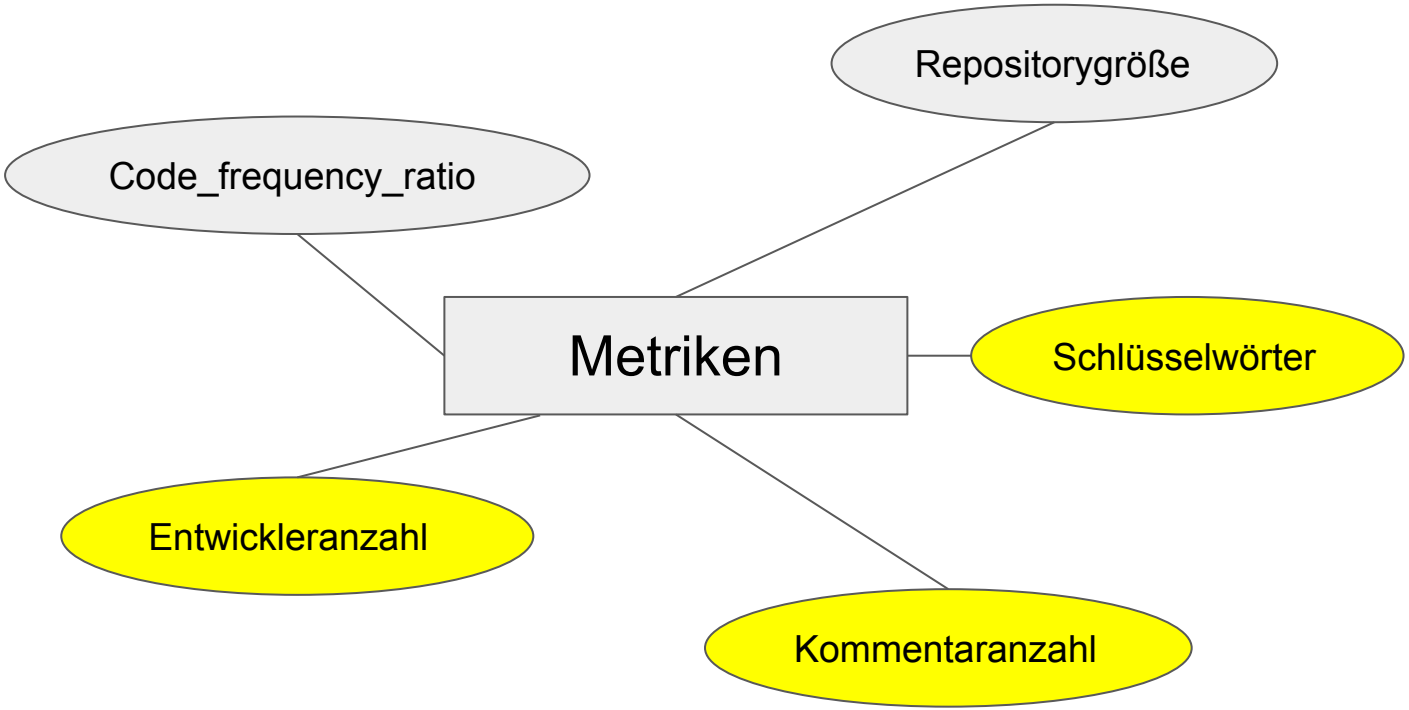
3. Ansätze und Methoden

3.2 Methoden



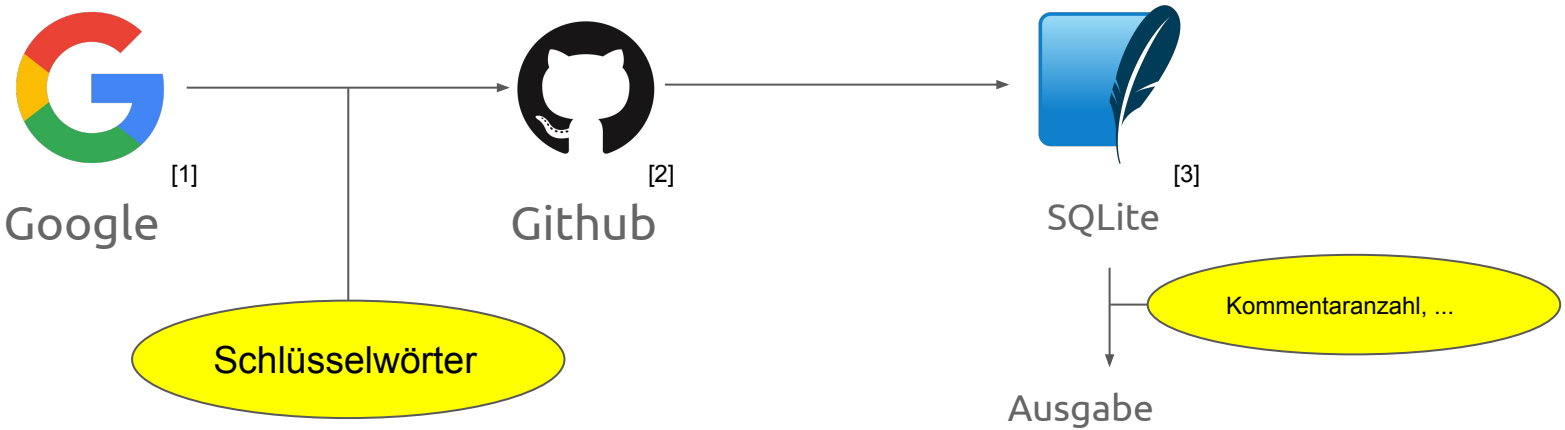
3. Ansätze und Methoden

3.2 Methoden



3. Ansätze und Methoden

3.1 Ansatz



4. Umgesetztes Tool

4.1 Dokumentation

Workflow

QIFTool is written in Python(3.8) and uses Google's Custom Search JSON API in conjunction with Google's Custom Search Engine to filter issues directly on Github. Keywords will be read out of a configuration file to determine which issues should be prefiltered. All prefiltered issues found by the engine will be inserted for research and caching purposes into a SQLite database. Afterwards it uses the official Github API (PyGithub) to look through the available pieces of metainformation inside each found issue and compares them to the other metrics set in the configuration file to only show the issues that match the requirements.

How to run the program

1. Download the `qiftool.py` and `requirements.txt` files from the repository
2. Place both files in the desired location
3. Open the terminal and navigate to the files' location
4. Install all dependencies by running `pip3 install -r /path/to/requirements.txt` or just `pip install -r /path/to/requirements.txt` depending on your python version
5. Run the program by using `python3 qiftool.py`
6. By running it for the very first time the tool should have created a `config.ini` file inside the tool's folder. Fill out the necessary parameters following the instructions in Configuration file
7. With the `config.file` filled out run the programm again just like in step 5
8. The tool should now operate properly and an interactive mode will be seen. Follow Interactive mode for further instructions

Configuration file

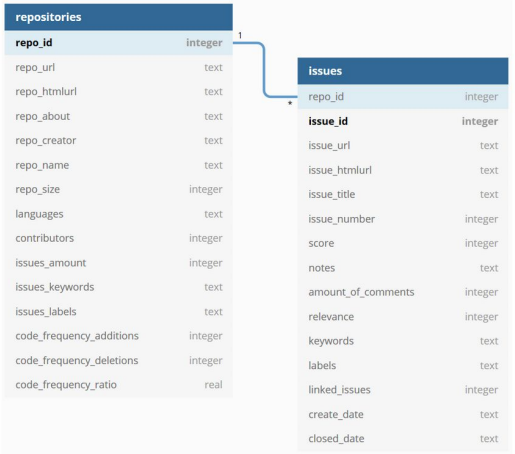
This file is created by running the program for the very first time. It is used to give the user a space to use their own parameters used by the tool. The file contains three sections for the user to fill out.

```
[DEFAULT]
  path_of_database = current
  path_of_download = current
[credentials]
  github_api_key = randomnumberandlettersinlowercase
  google_api_key = randomlettersinuppderandlowercaseandsymbols
  google_cse_id = randomleterandnumbersinlowercase
[metrics]
  keywords = technical debt refactor rewrite
  issue_comments = 5
  repo_contributors = 50
```

1. `[DEFAULT]` this section contains the path for the database and downloaded repositories to be stored in. The user is able to create their own path with the location of the 'qiftool.py''s as a pivot. These can be changed by providiung a valid path on your machine.
2. `[credentials]` this section contains the corresponding credentials necessary to run the used APIs
   i. `[github_api_key]`
      a. register on github
      b. use this link and click on 'generate new token' to create a new key
      c. paste the key as a parameter
   ii. `[google_api_key]`

SQLite - Database

This tool uses the SQLite version 3.33.0 (2020-8-14) library. The database created with this tool consists of two tables with one table refering to the other in a 1:n relation.



Repositories

Attribute	Datatype	Description
`repo_id (primary key)`	`integer`	indentifier for a repository
`repo_url`	`text`	url for the JSON file of this repository
`repo_htmlurl`	`text`	url that refers to the web based github repository
`repo_about`	`text`	text & contains the 'about' of the repository found on github. It's a brief description of the repository
`repo_creator`	`text`	name of the creator of this repository or fork
`repo_name`	`text`	name of the repository
`repo_size`	`integer`	size of this repository in MB
`languages`	`text`	list of programming languages used in this repository
`contributors`	`integer`	amount of contributors of this repository

4. Umgesetztes Tool

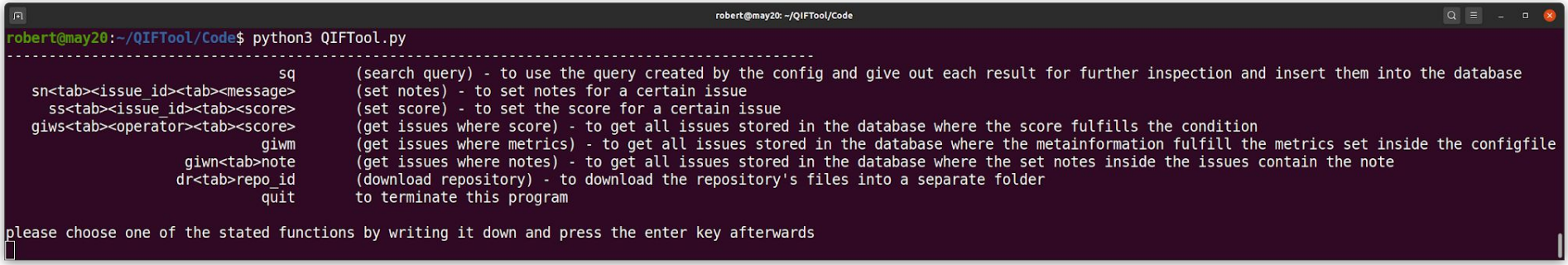
4.2 Konfigurationsdatei

```
[DEFAULT]
path_of_database = current
path_of_download = current

[credentials]
github_api_key = randomnumberandlettersinlowercase
google_api_key = randomlettersinuppderandlowercaseandsymbols
google_cse_id = randomleterandnumbersinlowercase

[metrics]
keywords = technical debt      refactor        rewrite
issue_comments = 5
repo_contributors = 50
```

.ini Tab Width: 8 Ln 13, Col 23 INS

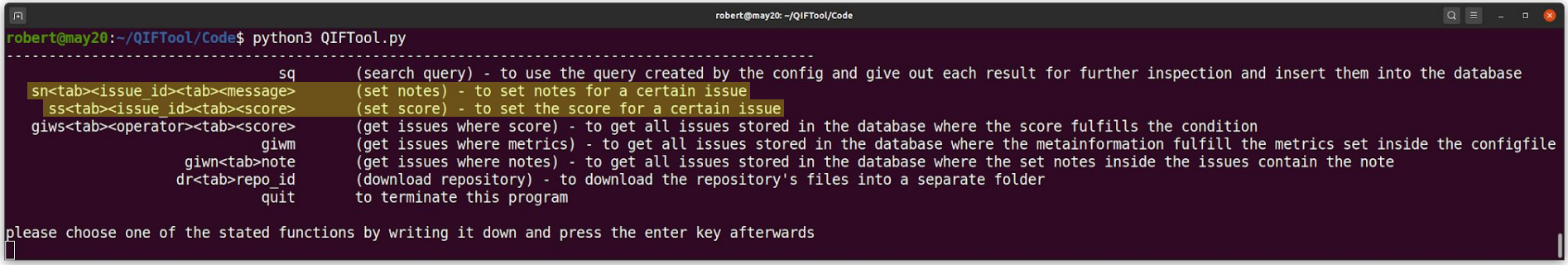
4. Umgesetztes Tool

4.3 Interaktiver Modus



4. Umgesetztes Tool

4.3 Interaktiver Modus



4. Umgesetztes Tool

4.3 Interaktiver Modus

```
please choose one of the stated functions by writing it down and press the enter key afterwards
sq
-------------------------------------------------------------------------------------------
                    Used Keywords: technical debt     refactor        rewrite
    Used minimum amount of comments: 5
 Used minimum amount of contributors: 50

-------------------------------------------------------------------------------------------
        repo_id: 8162715
      repo_name: saleor
     repo_about: A modular, high performance, headless e-commerce storefront built with Python, GraphQL, Django, and ReactJS.
       html_url: https://github.com/mirumee/saleor/issues/3915
       issue_id: 428807004
    issue_title: Refactor test fixtures
          score: 0
      relevance: 2
   linked_issues:
          notes:
-------------------------------------------------------------------------------------------
               Is refactoring or rewriting the most efficient and cost-effective way to realize a high-quality and robust codebase for NSS? 36
-------------------------------------------------------------------------------------------
        repo_id: 24306004
      repo_name: node-solid-server
     repo_about: Solid server on top of the file-system in NodeJS
       html_url: https://github.com/solid/node-solid-server/issues/788
       issue_id: 362373746
    issue_title: Is refactoring or rewriting the most efficient and cost-effective way to realize a high-quality and robust codebase for NSS?
          score: 0
      relevance: 3
   linked_issues:
          notes:
-------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------
        repo_id: 6093316
      repo_name: DefinitelyTyped
     repo_about: The repository for high quality TypeScript type definitions.
```

4. Umgesetztes Tool

4.3 Interaktiver Modus

```
please choose one of the stated functions by writing it down and press the enter key afterwards
sq
-------------------------------------------------------------------------------------------
                    Used Keywords: technical debt     refactor        rewrite
    Used minimum amount of comments: 5
 Used minimum amount of contributors: 50

-------------------------------------------------------------------------------------------
        repo_id: 8162715
      repo_name: saleor
     repo_about: A modular, high performance, headless e-commerce storefront built with Python, GraphQL, Django, and ReactJS.
       html_url: https://github.com/mirumee/saleor/issues/3915
       issue_id: 428807004
    issue_title: Refactor test fixtures
          score: 0
      relevance: 2
   linked_issues:
          notes:
-------------------------------------------------------------------------------------------
               Is refactoring or rewriting the most efficient and cost-effective way to realize a high-quality and robust codebase for NSS? 36
-------------------------------------------------------------------------------------------
        repo_id: 24306004
      repo_name: node-solid-server
     repo_about: Solid server on top of the file-system in NodeJS
       html_url: https://github.com/solid/node-solid-server/issues/788
       issue_id: 362373746
    issue_title: Is refactoring or rewriting the most efficient and cost-effective way to realize a high-quality and robust codebase for NSS?
          score: 0
      relevance: 3
   linked_issues:
          notes:
-------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------
        repo_id: 6093316
      repo_name: DefinitelyTyped
     repo_about: The repository for high quality TypeScript type definitions.
```

5. Fazit und Ausblick

5.1 Fazit

- Erfüllt seinen Zweck
- “Schlüsselwörter” als effektivste Metrik

5. Fazit und Ausblick

5.2 Ausblick

- Erweiterungsmöglichkeiten
- Multithematisch
- Grundbaustein für weitere Forschung

Vielen Dank für Ihre Aufmerksamkeit!

Bildquellen

[1] Google: https://upload.wikimedia.org/wikipedia/commons/thumb/5/53/Google_%22G%22_Logo.svg/512px-Google_%22G%22_Logo.svg.png

[2] Github: https://github.githubassets.com/images/modules/logos_page/GitHub-Mark.png

[3] Sqlite: https://upload.wikimedia.org/wikipedia/commons/thumb/9/97/Sqlite-square-icon.svg/1200px-Sqlite-square-icon.svg.png