An Exploratory Study of Documentation Strategies for Product Features in Popular GitHub Projects

ICSME 2022

Tim Puhlfürß
@TimPuhl

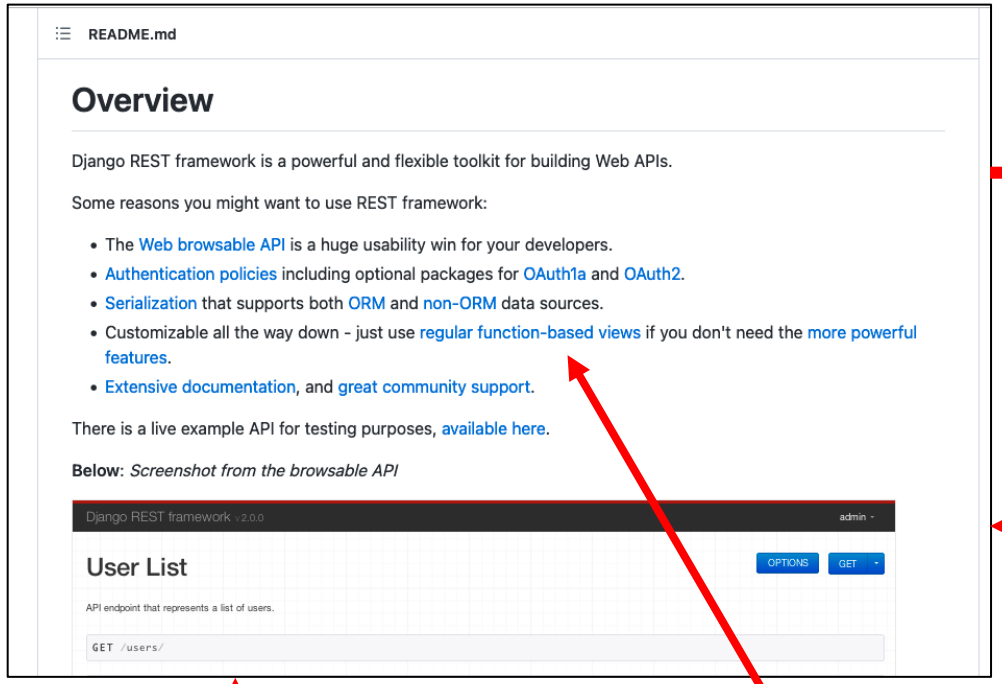
Lloyd Montgomery
@LloydThinks

Walid Maalej
@maalejw

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

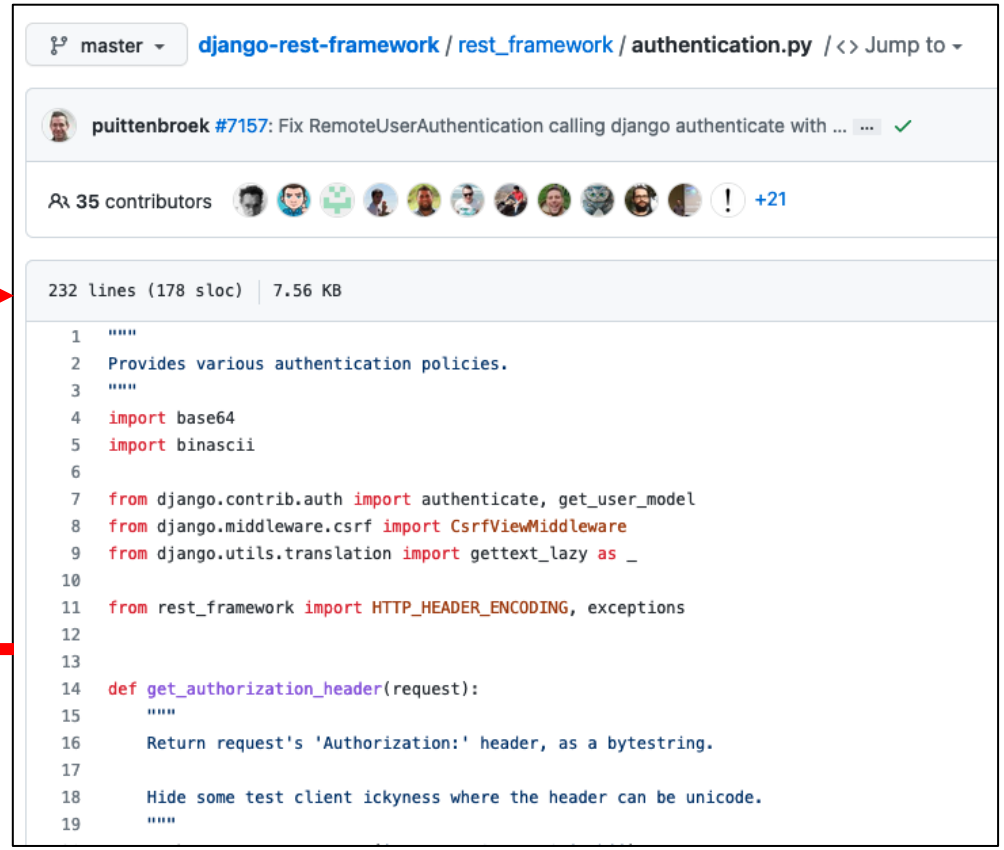
DASHH Data Science in Hamburg
HELMHOLTZ Graduate School for the Structure of Matter

Research Objective



(RQ2a) Connection?

(RQ2b) Connection?



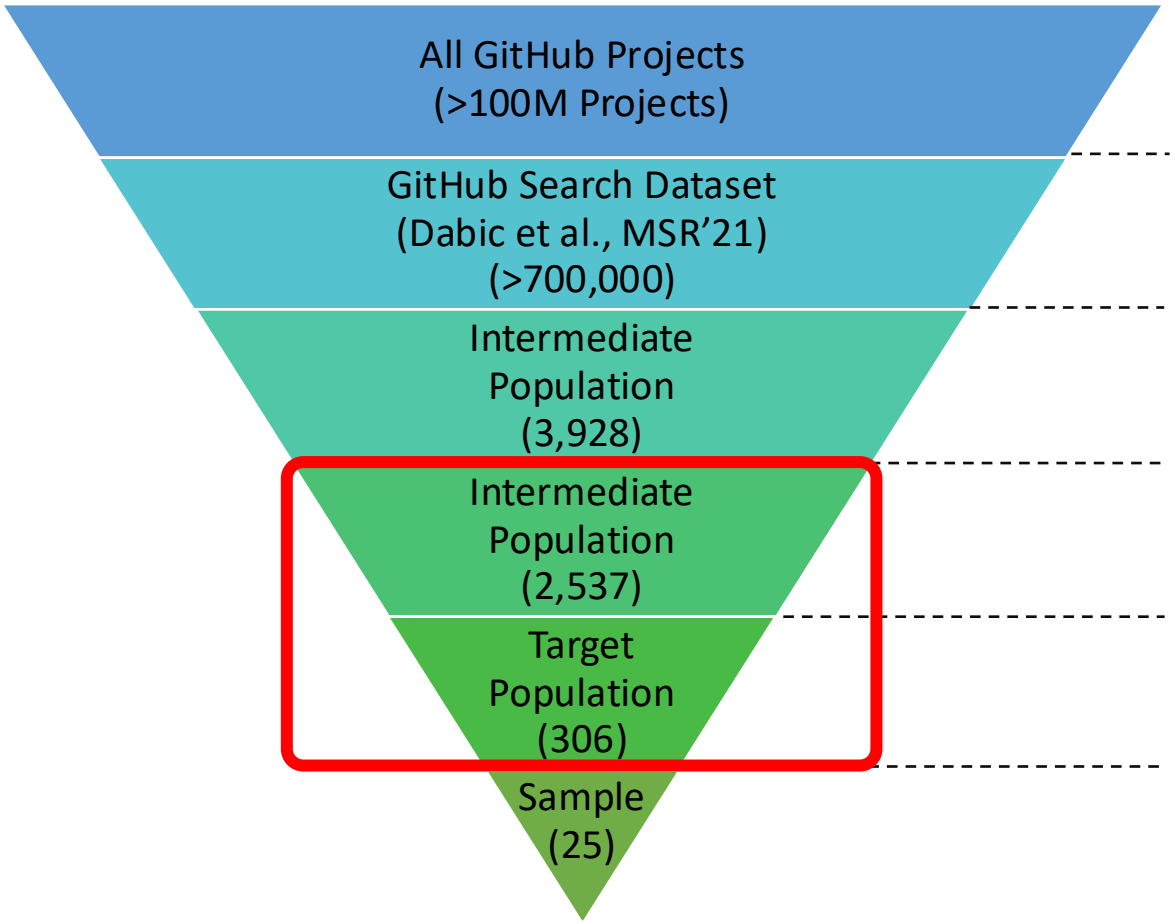
Descriptive elements? **(RQ1b)**

Textual artefacts to document product features? **(RQ1a)**

Research Method: In-depth Content Analysis of 25 GitHub Projects

Methodology – Population and Sampling

Scope: Very Popular Projects with Documentation of High Quality



Filter by Dabic et al.:
≥10 Stars

Filter on Dataset:
≥ 4,000 Stars; Not a Fork, Not Archived

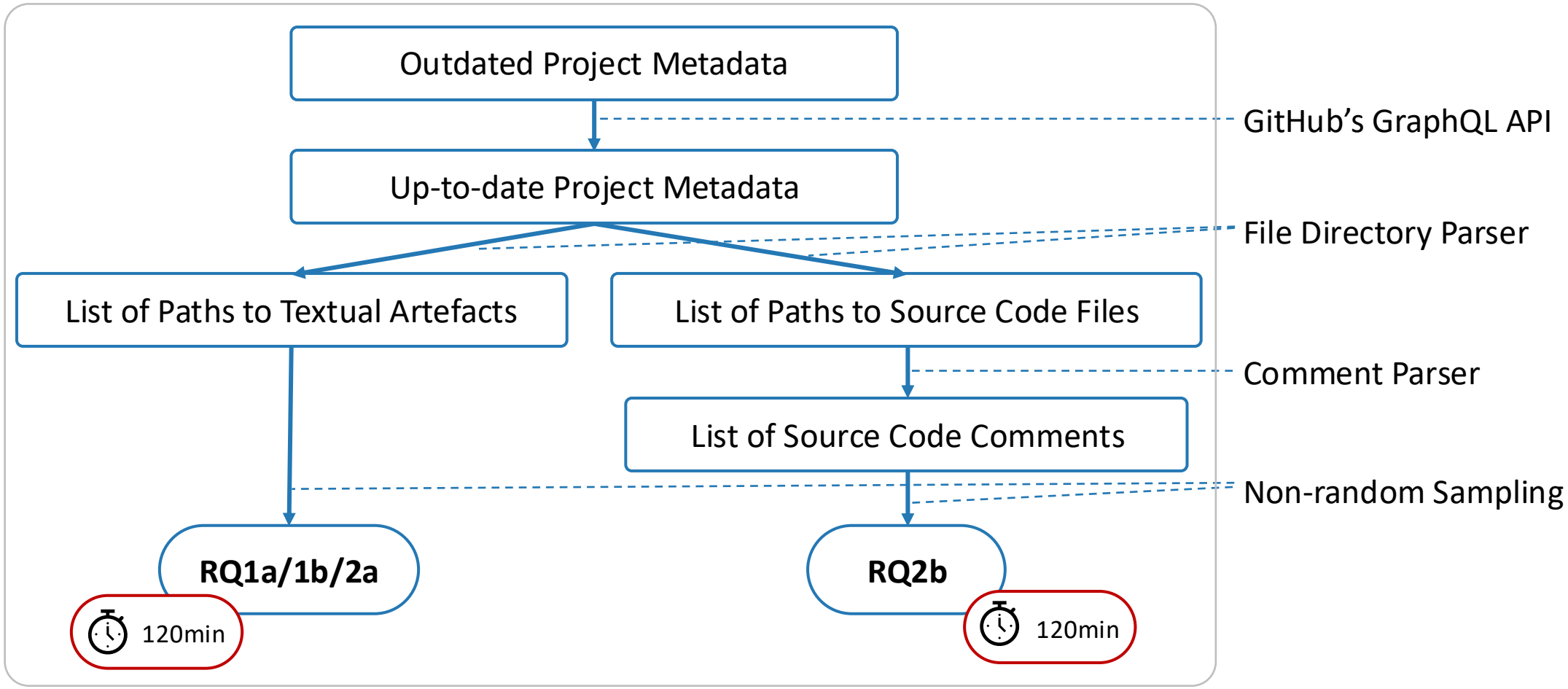
Filter via GitHub's GraphQL API:
≥ 5,000 Stars; Last Commit ≥ June 3, 2020;
Not a Fork, Not Archived / Disabled / Locked / Private

Filter via Beautiful Soup (Website Scraper):
Fulfillment of GitHub's *Community Profile* Indicator

Random Stratified Sampling:
Strata Characteristic: Primary Programming Language

Methodology – Analysis Approach

Per Project

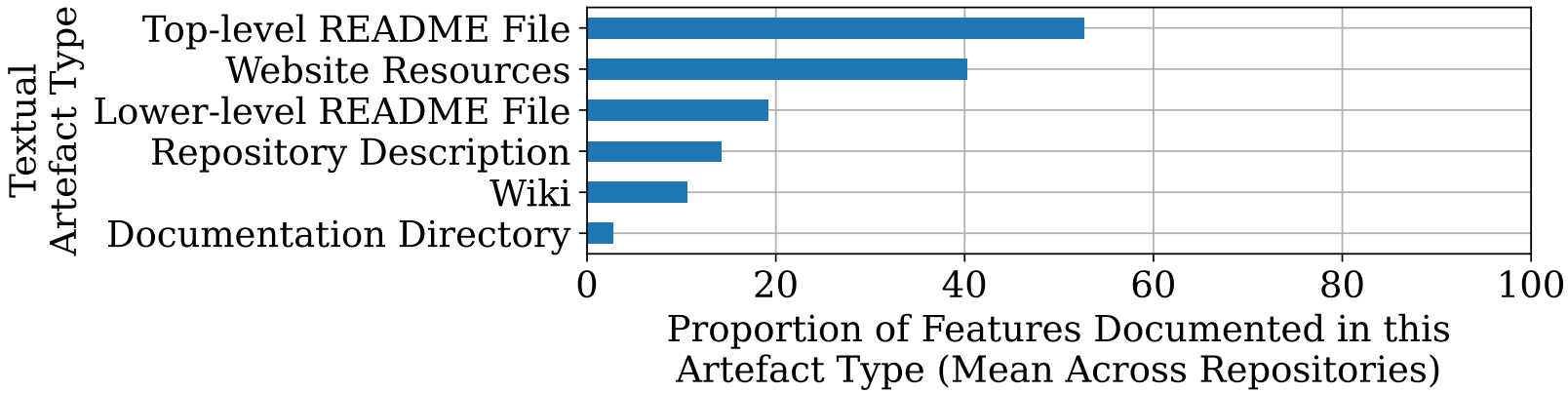


RQ1a: Textual Artefacts Used to Document Product Features

„**Features** [...]

– Full Navigation Control: Swiper comes with all required built-in navigation elements, such as Pagination, Navigation arrows and Scrollbar“

Top-level README File of *swiper*



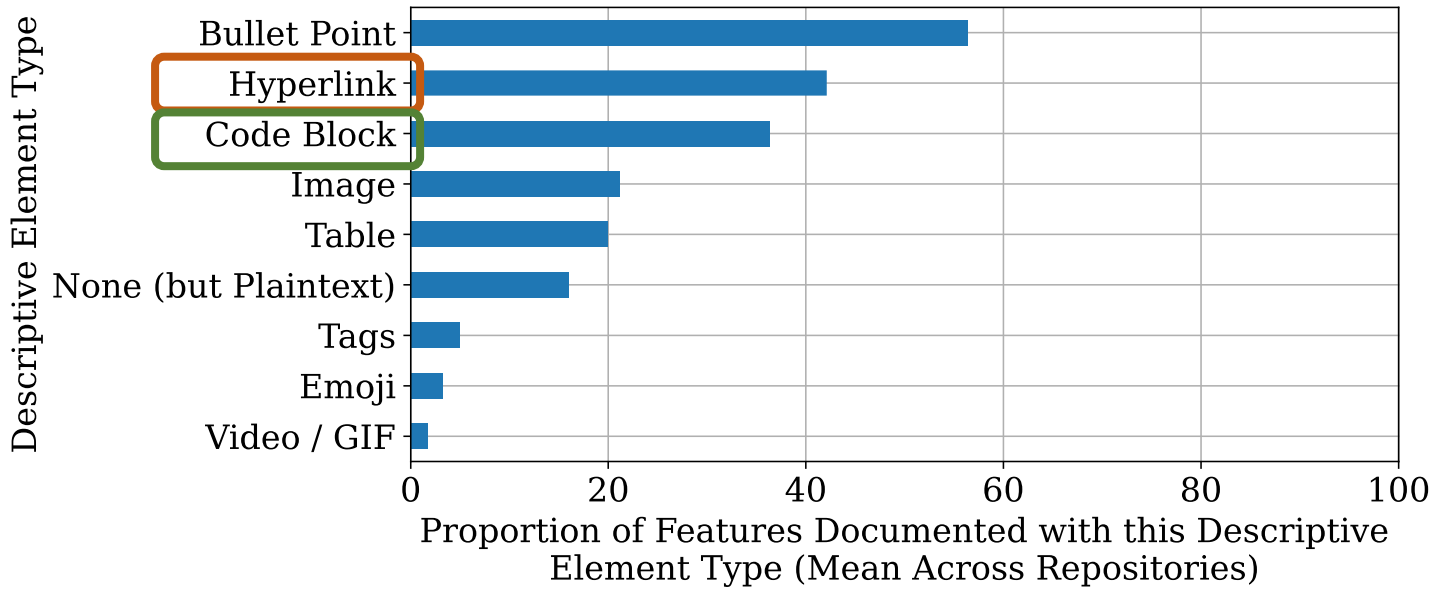
RQ1a: Textual Artefacts Used to Document Product Features

Usage of GitHub-supported documentation means: relatively rare
(in comparison to the usage of website resources)

Recommendation: Use beneficial combinations of textual artefacts.

E.g., top- and lower-level README files

RQ1b: Descriptive Elements Used to Document Product Features



Website Resources of *django-rest-framework*

RQ1b: Descriptive Elements Used to Document Product Features

Features often “entangled” in unstructured text

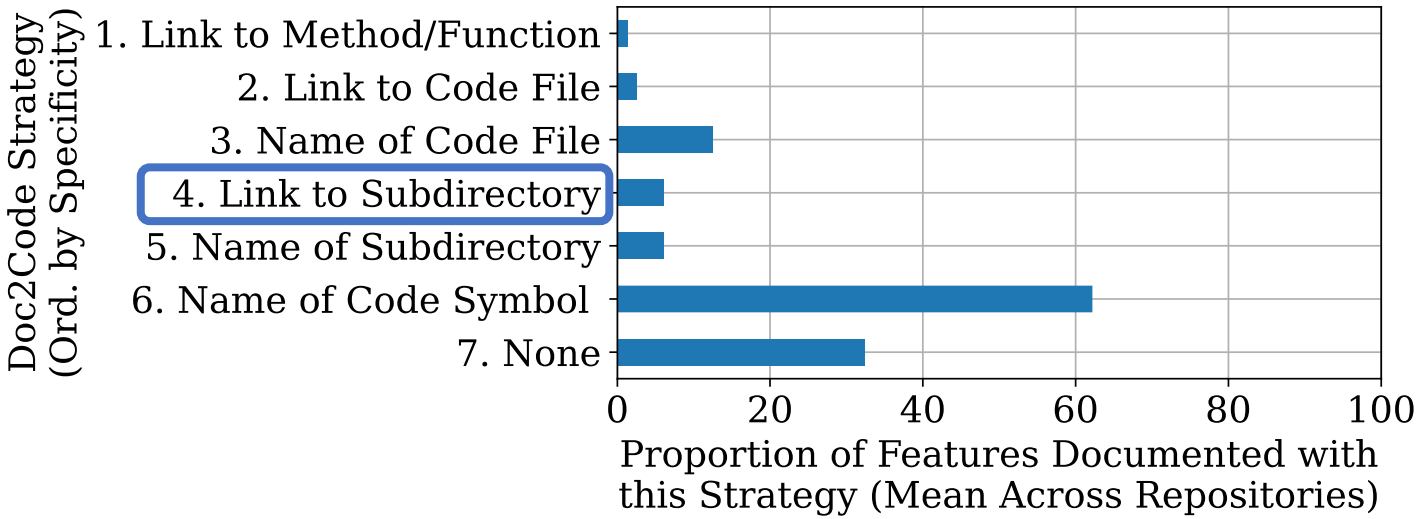
Recommendation: Provide a structured presentation of product features / functional requirements.

E.g., use tables, bullet points, and emojis to improve structure

RQ2a: Strategies to Link from Textual Artefacts to Source Code

„The code is set up into several main directories: [...]
- zoo: contains code to directly download and use pretrained models from our model zoo“

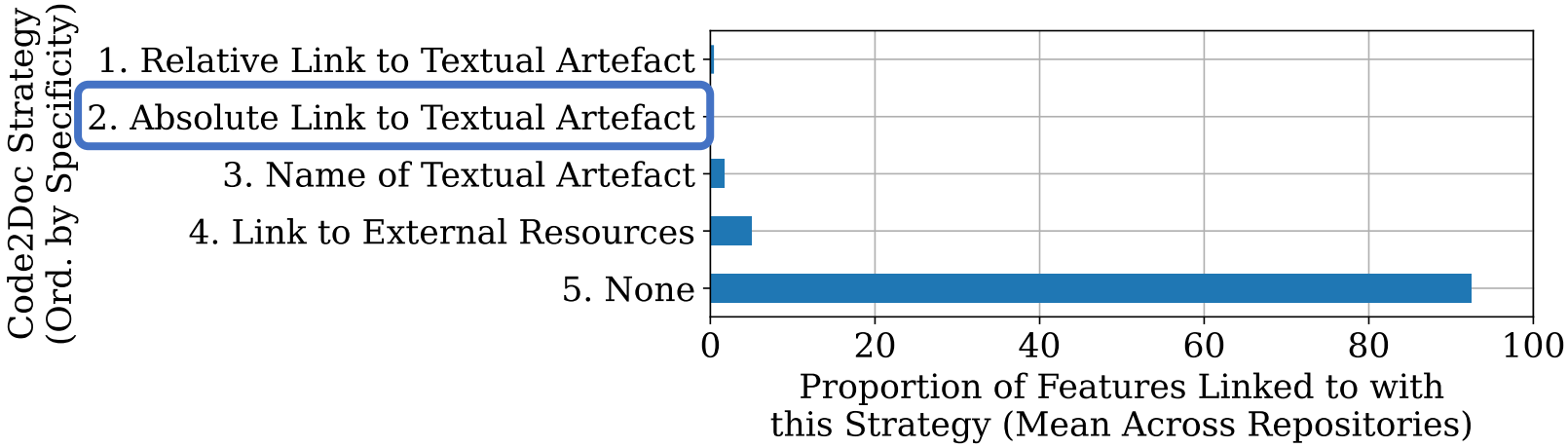
Top-level README file of *ParlAI*



RQ2b: Strategies to Link from Source Code to Textual Artefacts

„ * Register logs source configuration, used by LogStream components
* @see https://github.com/elastic/kibana/blob/master/x-pack/plugins/infra/public/components/log_stream/log_stream.stories.mdx#with-a-source-configuration“

Comment in a code file of *kibana*



RQ2a/b: Strategies to Link between Textual Artefacts and Source Code

Sparse usage of tracing strategies to link textual artefacts and code

Recommendation:
Improve tracing information available in textual artefacts and source code files

(1) Hyperlinks → direct navigation between files
(2) Use documentation-focused Tools

An Exploratory Study of Documentation Strategies for Product Features in Popular GitHub Projects

Tim Puhlfürß
@TimPuhl
tim.puhlfuerss@uni-hamburg.de

Lloyd Montgomery
@LloydThinks
lloyd.montgomery@uni-hamburg.de

Walid Maalej
@maalejw
walid.maalej@uni-hamburg.de

Preprint:
https://doi.org/10.48550/arXiv.2208.01317

Artefacts:
https://doi.org/10.5281/zenodo.7107674

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

DASHH
Data Science in Hamburg
HELMHOLTZ Graduate School for the Structure of Matter