Contextual Documentation Referencing on Stack Overflow

Sebastian Baltes, Christoph Treude, Martin P. Robillard

Abstract—Software engineering is knowledge-intensive and requires software developers to continually search for knowledge, often on community question answering platforms such as Stack Overflow. Such information sharing platforms do not exist in isolation, and part of the evidence that they exist in a broader software documentation ecosystem is the common presence of hyperlinks to other documentation resources found in forum posts. With the goal of helping to improve the information diffusion between Stack Overflow and other documentation resources, we conducted a study to answer the question of how and why documentation is referenced in Stack Overflow threads. We sampled and classified 759 links from two different domains, regular expressions and Android development, to qualitatively and quantitatively analyze the links' context and purpose, including attribution, awareness, and recommendations. We found that links on Stack Overflow serve a wide range of distinct purposes, ranging from citation links attributing content copied into Stack Overflow, over links clarifying concepts using Wikipedia pages, to recommendations of software components and resources for background reading. This purpose spectrum has major corollaries, including our observation that links to documentation resources are a reflection of the information needs typical to a technology domain. We contribute a framework and method to analyze the context and purpose of Stack Overflow links, a public dataset of annotated links, and a description of five major observations about linking practices on Stack Overflow. Those observations include the above-mentioned purpose spectrum, its interplay with documentation resources and applications domains, and the fact that links on Stack Overflow often lack context in form of accompanying quotes or summaries. We further point to potential tool support to enhance the information diffusion between Stack Overflow and other documentation resources.

Index Terms—Community Question Answering, Software Documentation, Information Diffusion, Hyperlinks, Stack Overflow

1 Introduction

The knowledge-intensive nature of current-day software engineering means that software developers are continually in search of knowledge. A popular model for knowledge sharing on the Internet is the community question answering site, with Stack Overflow [1] serving as the de facto forum for most programmers [2]. On Stack Overflow, registered users can post questions, answer posted questions, and comment on questions and answers by other users, which can then be viewed by anyone. As of December 2019, Stack Overflow archives 19M questions, 28M answers, and 72M comments. At this scale, Stack Overflow constitutes a major information broker between posters, contributors, and non-contributing readers (so-called "lurkers").

Stack Overflow, however, does not exist in isolation—the site is only one of many sources of programmer knowledge in a software documentation ecosystem. Past research has extensively characterized the strengths and weaknesses of Stack Overflow (e.g., good at "how-to" documentation [3], bad at completeness [4]) compared to other sources, such as API documentation (e.g., good at structure [5], bad at scenarios [6]). Meng et al.'s observational study corroborates that developers seek a diversity of documentation content when solving programming tasks [7].

- *Sebastian Baltes is with The University of Adelaide, Australia. E-mail: sebastian.baltes@adelaide.edu.au*
- *Christoph Treude is with The University of Adelaide, Australia. E-mail: christoph.treude@adelaide.edu.au*
- *Martin P. Robillard is with McGill University, Montral, Canada. E-mail: martin@cs.mcgill.ca*

Manuscript received March 11, 2019.

With these complementary strengths and weaknesses, it is only natural that links exist from one source to another. In fact, previous studies found that link sharing is a significant phenomenon on Stack Overflow that make the site part of a larger interconnected network of online resources used and referenced by developers [8], [9]. Given the crucial role that on-line resources play in developers' quest for technical knowledge, it is important to know how information is diffused between resources types so we can facilitate this quest (see Section 2).

We conducted a multi-case study to answer the question of *how and why documentation is referenced in Stack Overflow threads*. We sampled 759 links from two different domains (Java regular expressions and Android development), classified and qualitatively analyzed them, and then used the resulting data to derive association rules and build logistic regression models to identify properties of Stack Overflow questions that attract links to documentation resources.

Our main findings include that links on Stack Overflow serve widely diverse purposes that range from simple pointers to API documentation over links to concept descriptions on Wikipedia to suggestions of software components and background readings. This *purpose spectrum* (see Section 7) allows us to modulate Stack Overflow's requirement to add context for links [10]. We also find that links to documentation resources are a reflection of the information needs typical to a technology domain, with significant differences between the two domains in our multi-case study.

Our main contributions are: (1) a framework and method to analyze the *context* and *purpose* of documentation links on Stack Overflow, (2) a public dataset with 759 annotated

links that other researchers can use, and (3) a description of five major observations about linking practices on Stack Overflow, with detailed links to evidence, implications, and a conceptual framework to capture the relations between the five observations.

The remainder of this paper is structured as follows: We provide additional background and motivation in Section 2 and outline our study design in Section 3. Section 4 describes our method for link sampling and classification, Sections 5 and 6 describe our qualitative and quantitative analyses, respectively. Section 7 presents the major findings derived from these analyses, Section 8 describes threats to validity. We conclude the paper in Section 9.

2 Background and Motivation

This work is a systematic investigation of current **information diffusion (link sharing) practices on Stack Overflow**, with the goal of informing the development of **advanced technology to facilitate this diffusion**. This research takes place in the context of **previous studies on information diffusion in on-line developer communities**.

Information Diffusion on Stack Overflow

Stack Overflow explicitly encourages the inclusion of links to external resources in answers, but requests that users add context so that "fellow users will have some idea what it is and why it's there." [10]. This advice is overly general. Not all link targets need to be quoted, and in some cases, the context for a link is obvious. However, deciding when and how to include links to other documentation sources in Stack Overflow posts requires differentiating common linking practices and understanding their unique characteristics. The following examples illustrate the richness and diversity of linking practices on Stack Overflow.

When considering the potential value of links on Stack Overflow, the best case scenario is the recommendation of specific information relevant to the thread (links are in bold):

> ...have a look at **Greedy, Reluctant, and Possessive Quantifiers** section of the Java RegEx tutorial... [11]

In this case, a contributor provided a comment to point the original poster to a section of a tutorial introducing the concept of regular expression quantifiers and explaining how to use them. These "ideal" links provide clear value added to the thread, and form a type of information that can even be automatically mined to improve information discovery [12].

However, the reality of linking practices goes broadly beyond this expected scenario. For example, links to obvious documentation resources can be introduced defensively by the original poster themselves, to avoid having a question *downvoted* [13]:

> I've already tried this solution (**http://developer.android.com/training/articles/security-ssl.html**) but I still have the same error:... [14]

Other links bind a reference to library classes to its documentation. This can be useful to help make code fragments more self-explanatory [15], but we observed that such links are also provided for well-known, pervasive classes:

> When you want to return more than one result, you need to return an array (String[]) or a Collection like an **ArrayList**, for example. [16]

From the point of view of links as mechanisms to increase the flow of valuable software development knowledge, degenerate practices include providing links to comic strips (such as xkcd) and similar sites:

> ...reminds me of this **xkcd** [17]

As these examples show, linking practices on Stack Overflow are diverse and the intrinsic value of a link as a carrier of relevant technical information is not uniform. The first example link, to a specific section of a tutorial, has an obvious purpose and value. The link to a comic strip is clearly noise. Between these extremes lies a gray zone where links play different roles in different contexts.

Enhancing Information Diffusion

As illustrated above, links in on-line developer forums can fulfill the important mission of complementing documentation with explanations of concepts or descriptions of code elements. However, a manual linking process is prone to omissions. A number of techniques have been proposed to automatically enhance on-line resources through linking and recommendation.

The idea of automatically enhancing information diffusion is clearly captured by Gao et al.'s proposal to automatically add links to recognized entities in Stack Overflow posts from a database of popular URLs, and taking into account the context in which the entity appears [18]. A different take on the problem is offered by Li et al. [12], who built a collaborative filtering recommender system to recommend other learning resources, based on co-occurrences of links in Stack Overflow posts. In approaches that are based on existing link data, the automatic linking system relies on the assumption that the underlying linking practices are sound. Our study sheds light on the linking practices that are used as foundations for collaborative filtering. Xu et al.'s deep-learning-based approach for predicting semantically linkable knowledge in developer forums [19] avoids the issue of relying on existing links. This is an important advancement for improving information diffusion in knowledge networks. However linking that is based on the semantics of the text may not necessarily take into account the *purpose* for linking a knowledge unit (e.g., the comic strip mentioned above). Our study focuses specifically on eliciting the purpose of links so that it is possible to account for it when enabling information diffusion though automated approaches.

Content is one concern for documentation ecosystems, but quality is another important one. Previous work has attempted to automatically identify high-quality posts using features based on the number of edits on a question [20], author popularity [21], and code readability [22]. In their conceptual framework of success factors for Stack Overflow questions, Calefato et al. [23] considered the presence of links as one aspect of a question's presentation quality. However, they did not find a significant effect of the fact that a question contained a link on the success of that question, that is whether it attracted an accepted answer. A direction

of future work is to consider not only the presence of a links, but also their purpose and targets, as enabled by our study.

Studies of Information Diffusion

There have been different studies investigating individual aspects of link usage on Stack Overflow. Gomez et al. [8] conducted a preliminary study of the links found on Stack Overflow. Their study focused on the different types of links in posts (not comments) and it did not factor in a distinction based on the domain. In this article, we investigate two specific domains, which allows us to understand the data in a specific context. Moreover, we integrate an analysis of the purpose of the information sharing that goes beyond a basic description of its nature.

Vincent et al. [9] analyzed the usage of Wikipedia by Stack Overflow authors. They found that 1.28% of all Stack Overflow posts contain links to Wikipedia. Using version 2018-07-31 of the *SOTorrent* dataset [24], we identified 1.94% of all threads, but only 0.85% of all posts, to contain links to Wikipedia. Also considering links in comments, which Vincent et al. did not, the ratio of threads with links to Wikipedia increases to 2.55%.

Ye et al. contributed a study of link sharing on Stack Overflow that focuses on the sharing of links to other Stack Overflow posts [25]. In contrast, our study covers links to all external resources, and for this reason an important part of our study design addresses the problem of categorizing the types of documentation referenced. The part of the Ye et al. study that is the most complementary to ours is their analysis of the purpose of links. However, because the study focuses on internal links, their classification does not include purposes that would be exclusive to resources outside of Stack Overflow itself. Their classification is also more abstract, with four categories of purpose (excluding the "other" category), whereas we analyze the link purposes at a finer granularity.

In addition to Stack Overflow, studies have also investigated linking practices in other context. Hata et al. [26] studied the role of links contained in source code comments in terms of prevalence, link targets, purposes, decay, and evolutionary aspects. They report that links can be fragile since link targets change frequently or disappear. Links are also shared as part of code review. Jiang et al. contribute a study of link sharing in review comments [27], reporting that roughly half the links they identified refer to resources outside the project. This observation further motivates our study in that the observations we make about information diffusion may also be applicable to contexts other than question and answer forums.

3 Study Design

To investigate *how* and *why* documentation resources are referenced in Stack Overflow threads, we conducted a mixed-methods study involving a qualitative analysis of 759 links from 742 different threads and a quantitative analysis using association rule mining and logistic regression models.

Research Questions

The overall goal of the study is to discover the roles that links to documentation play in Stack Overflow threads and thus pave the way for a more systematic treatment of documentation references on Q&A sites for software developers. We split our research questions into two sub-questions:

RQ1 What is the **context** around documentation links in Stack Overflow threads? With this question we study *how* links are provided.

RQ2 What is the **purpose** that documentation links in Stack Overflow threads serve? With this question we study *why* links are provided.

With these questions, our aim was to collect specific insights about linking practices on Stack Overflow, that can support actionable implications for authors and readers of Q&A forums and for the development of technology based on the analysis of such forums.

Our first research question was motivated by the fact that Stack Overflow encourages users to provide **context** for links [10], in particular by quoting external sources [28]. We qualitatively analyzed whether users follow this advice (see Section 5), but we also built logistic regression models capturing different features of Stack Overflow posts to quantitatively analyze which of those features are related to the presence of documentation links (see Section 6).

As the examples in Section 2 illustrate, links on Stack Overflow serve diverse **purposes**. To conduct a structured analysis of those purposes, we first built a classifier that was able to identify links to the most frequently referenced documentation resources (see Section 4). Based on a stratified sample of documentation links identified using the classifier, all three authors independently coded the purpose of 759 links using a jointly developed coding guide (see Section 5). We mined the resulting data for association rules between documentation resources and assigned purposes and then used our qualitative and quantitative results to corroborate five major findings about linking practices on Stack Overflow (see Section 7).

Cases Studied: Regex and Android

Because even a cursory inspection of Stack Overflow threads shows clear differences in the use of references to external documentation, we structured our research as a multi-case study of linking practices for two different domains: use of regular expressions in Java (*Regex*), and Android development (*Android*). We bounded our investigation to clearly-defined domains to support a richer analysis of linking practices in the context of the wider documentation ecosystem they integrate. We selected *Regex* and *Android* because they constituted two very different domains (library vs. framework, small vs. large, integrated in the programming language vs. third-party, theoretically vs. practically grounded), and because we were familiar with both technologies. The importance of this latter aspect is not to be underestimated as a contributor to the meaningfulness of qualitative data analysis.

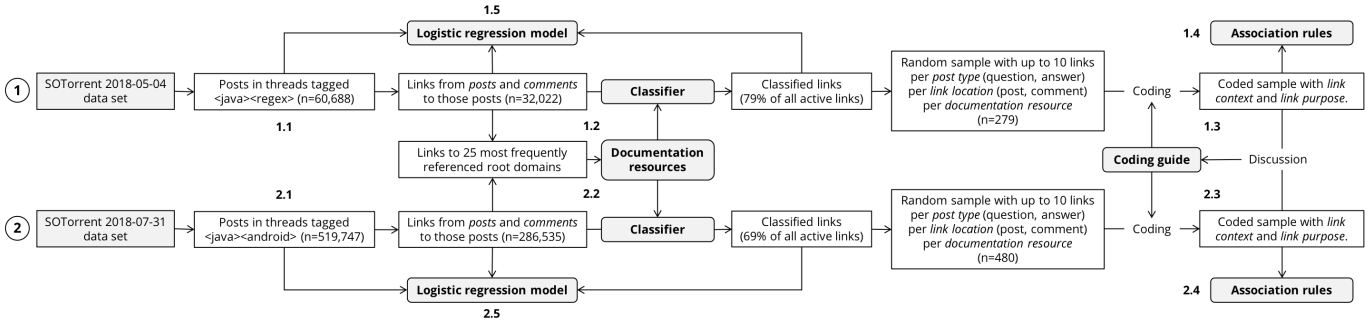

Fig. 1. Study process for both cases (1: *Regex* and 2: *Android*). The two cases were studied sequentially.

Overview of the Research Process

Despite the ready availability of structured data from Stack Overflow, generating reliable insights about linking practices requires an extensive combination of analytical processing and manual inspection. Figure 1 outlines the general process we followed. The research proceeded sequentially: we first completed an entire iteration for *Regex* (referenced as number 1 on the figure), and then repeated the process for the second case (*Android*), referenced as number 2.

In the following description, numbers refer to the step in the process overview (indicated after the period in Figure 1).

The first step was to retrieve all Stack Overflow threads related to each case (N.1). For this purpose we utilized the *SOTorrent* dataset [24]. For the *Regex* case, we retrieved all threads with tags java and regex, and for the *Android* case, the threads with tags java and android. For each case, we used the most recent release at the time (2018-05-04 for the *Regex* case [29] and 2018-07-31 for the *Android* case [30]).

The second step was to process the links to determine what they were linking to, and to abstract the target of the links to one of a small set of *documentation resource* categories (e.g., links to other Stack Overflow threads vs. links to API documentation). We built a URL mapper to classify links to such documentation resources using the 25 most frequently referenced root domains for each case (Section 4 and N.2 in Figure 1).

The classification of links was necessary to create a stratified sample for detailed analysis, i.e., a sample guaranteed to contain links to all different types of resources. The third step was then to draw samples containing links to all identified documentation resources and qualitatively analyze their context and purpose (see Section 5 and N.3 in Figure 1). This step involved extensive manual inspection and labeling of links in their context.

In step four, to investigate the motivation behind linking to documentation resources of a certain type, we used association rule mining [31] to investigate the relationship between resource type and purpose (see Section 6 and N.4 in Figure 1).

Finally, we built logistic regression models to analyze which properties capturing the question context attract links to documentation resources in comments and answers (see Section 6 and N.5 in Figure 1). In these models, we treat question features as independent variables and the presence of a link to a particular resource as dependent variable.

TABLE 1
Five most frequently referenced root domains and assigned documentation resources in *Regex*; the second column lists the number of posts referring to the corresponding domain as well as the frequency relative to all posts containing links ($n_{\text{posts}} = 21,758$).

Domain	#Posts (%)	Resource Categories
stackoverflow.com	5,120 (23.5%)	*StackOverflow*, *NotDocumentation*
regex101.com	4,439 (20.4%)	*NotDocumentation*
oracle.com	4,316 (19.8%)	*JavaAPI*, *JavaReference*, *OtherForum*
ideone.com	1,933 (8.9%)	*NotDocumentation*
regular-expressions.info	1,868 (8.6%)	*IndependentTutorial*

TABLE 2
Five most frequently referenced root domains and assigned documentation resources in *Android*; the second column lists the number of posts referring to the corresponding domain as well as the frequency relative to all posts containing links ($n_{\text{posts}} = 177,784$).

Domain	#Posts (%)	Resource Categories
stackoverflow.com	57,461 (32.3%)	*StackOverflow*, *NotDocumentation*
android.com	42,199 (23.7%)	*AndroidAPI*, *AndroidReference*
imgur.com	22,339 (12.6%)	*NotDocumentation*
github.com	18,259 (10.3%)	*OtherReference*, *NotDocumentation*
google.com	11,924 (6.7%)	*AndroidIssue*, *AndroidReference*, *OtherReference*, *OtherForum*

Replication Package

To support the complete replicability of this process and the verification of the results presented in this paper, we provide our coding guide, samples, and the analysis and data retrieval scripts as supplementary material [32].

4 Link Classification and Sampling

Links on Stack Overflow may point to resources other than documentation, e.g., tools or images. To be able to study links to documentation resources on Stack Overflow, we built a URL-based classifier that takes as input a link and outputs either one of 12 documentation resource categories that best describes the target of the link, or marks the link as *NotDocumentation* (see Table 3). Those 12 categories emerged during an iterative analysis of the most frequently linked domains. We used the classifier to categorize all links in the

TABLE 3
Assigned documentation resources for links in Stack Overflow posts and comments (NotDocumentation: links that our URL mapper classified as not pointing to a documentation resource, NotClassified: links that our URL mapper could not classify, DeadOrInvalid: links that were either unavailable or invalid).

Resource Category	#Links in *Regex*		#Links in *Android*	
All	35,022	(100.0%)	286,535	(100.0%)
Classified	25,917	(74.0%)	185,857	(64.9%)
Documentation	**15,430**	(44.1%)	**150,630**	(52.6%)
NotDocumentation	10,487	(29.9%)	35,227	(12.3%)
NotClassified	7,115	(20.3%)	83,989	(29.3%)
InvalidOrDead	1,990	(5.7%)	16,689	(5.8%)
Documentation	**15,430**	(100.0%)	**150,630**	(100.0%)
StackOverflow	5,656	(36.7%)	64,610	(42.9%)
JavaAPI	5,093	(33.0%)	7,403	(4.9%)
IndependentTutorial	2,419	(15.7%)	6,600	(4.4%)
JavaReference	957	(6.2%)	3,860	(2.6%)
Wikipedia	787	(5.1%)	5,218	(3.5%)
OtherAPI	253	(1.6%)	644	(0.4%)
OtherReference	262	(1.7%)	6,514	(4.3%)
OtherForum	3	(0.0%)	549	(0.4%)
AndroidAPI	N/A	(0.0%)	28,690	(19.0%)
AndroidReference	N/A	(0.0%)	23,421	(15.5%)
AndroidIssue	N/A	(0.0%)	1,301	(0.9%)
YouTube	N/A	(0.0%)	1,820	(1.2%)

two cases and then sampled links from each category of documentation links for our qualitative analysis.

Building the Classifier

As mentioned above, we built the link categorization and corresponding classifier following a grounded, iterative approach.

First, we ranked all referenced *root domains* according to the number of posts in which they were referenced (the root domain of en.wikipedia.org, for example, is wikipedia.org). Starting with the most frequently referenced root domain, we inspected the extracted links and either decided that they form a new resource category or assigned them to an existing one. Tables 1 and 2 show the five most frequently referenced root domains, meaning that those were the first five domains we derived resource categories from.

For each of the analyzed domains, we started by investigating the different paths that were linked from the Stack Overflow posts retrieved for the particular case. For both cases, the most frequent link target was the platform itself. Because such links are internal to platform, we created a dedicated documentation resource category *StackOverflow*. However, we soon realized that not all links to stackoverflow.com can be considered software documentation links, because the linked paths included user profiles (e.g., /users/1974143) or internal help pages (e.g., /help/mcve).

Instead of excluding the paths that we did not consider documentation targets, we followed a whitelisting approach. We first built regular expressions matching the paths of the domains that we identified as pointing to documentation resources (e.g., another Stack Overflow post). After integrating those regular expressions in our link classifier, we executed the classification and analyzed the links to the current domain that had not been classified yet. We then refined the regular expressions and repeated the process until all links to documentation resources were classified either as *Documentation*, *NotDocumentation*, or *InvalidOrDead* (see also Table 3). This process was performed by two authors who continuously discussed the emerging resource categories and associated regular expressions. All decisions in the process were made unanimously. The source code of the classifier, including the regular expressions for all documentation resources, is available on GitHub[1] and archived on Zenodo [33].

To conclude the above example, for the stackoverflow.com root domain we decided to only match links to questions, answers, post revisions, and comments—but not links to user profiles or pages with tips on how to write questions and answers (see above). To illustrate this classification approach, we briefly describe the path matching for this domain. As mentioned above, we modeled internal links as a separate documentation resource. The regular expressions for the corresponding *StackOverflow* documentation resource all start with:

```
^https?://((www|pt|ru|es)\\.)?stackoverflow\\.com
```

This prefix is followed by expressions matching the different paths we determined to point to documentation resources:

```
/(a|q|questions)/[\\d]+.*
/revisions.*
/posts/\\d+/revisions.*
/posts/comments.*
```

All other paths for the root domain stackoverflow.com are automatically classified as NotDocumentation. Root domains that we have not analyzed yet are automatically labeled as NotClassified, root domains that our tool determined to be invalid or dead are automatically classified as InvalidOrDead. This allowed us to track our progress. We continued with the next root domain once we could not find paths anymore that were incorrectly labeled as NotDocumentation.

We repeated the classification process for the 25 most frequently referenced root domains in both samples, which enabled us to classify 78.5% of all *active links* in the *Regex* sample and 68.9% of all active links in the *Android* sample. The ratio of classified active links can be derived from the data in Table 3 as follows: $^{(\text{Documentation}+\text{NotDocumentation})}/_{(\text{All}-\text{Dead})}$. Because we conducted our analysis of the *Android* case after the *Regex* case had been completed, the classifier for *Android* links was built by extending the preliminary *Regex* link classifier. Note that, as a last step, we re-ran the final classifier for the *Regex* case.

Table 3 shows the documentation resources we extracted for both cases. In the following, we briefly describe which kinds of documentation resources we assigned to the different categories together with exemplary links.

StackOverflow: This documentation resource consists of Stack Overflow questions, answers, post revisions, and comments (see details above).

OtherForum: We used this category to capture links to non-Stack-Overflow forum posts or threads including certain subpages of forums.oracle.com and groups.google.com.

{Java|Android|Other}{Reference|API}: The resource category *JavaReference* represents official Java documentation except for the Java API documentation, which is represented by *JavaAPI*. *OtherReference*, *AndroidReference*, *Other-*

1. https://github.com/sbaltes/condor

API, and *AndroidAPI* are analogously defined. Examples for *OtherAPI* include API documentation hosted on jsoup.org, commons.apache.org, and developers.facebook.com. Examples for *OtherReference* includes the cookbook subpages of jsoup.org, certain reports on unicode.org, and different GitHub Pages.[2]

AndroidIssue: Since Android issue descriptions were quite frequently referenced in the *Android* case, we created a dedicated category for them. Those links typically point to subpages of issuetracker.google.com or code.google.com/p/android/issues/.

IndependentTutorial: Links in this category point to independent tutorials. By 'independent', we mean tutorials not provided by authoritative entities such as Oracle for *Java* or Google for *Android*. Examples include tutorialspoint.com, mkyong.com, and rexegg.com.

Wikipedia: We assigned links to Wikipedia pages in various languages to this category.

YouTube: Especially in the *Android* case, Stack Overflow users frequently referenced YouTube videos. We assigned such links to this category.

Section 6 provides further examples for specific documentation resources, together with associated purposes we identified. Table 1 lists the five most frequently referenced root domains for *Regex*, together with the number of links to those domains and the assigned resource categories. Table 2 lists this information for *Android*.

Sampling

Because of the high effort involved in reviewing each link manually, we produced a sample of links to documentation resources for the qualitative analysis. We randomly sampled (up to) 40 links per documentation resource: We selected 20 links from questions (10 from question posts and 10 from question comments) and 20 links from answers (10 from answer posts and 10 from answer comments). Because some documentation resources had insufficient links to fulfill all of those selection constraints, the *Regex* sample contained only 279 links (and not $8 \cdot 40 = 320$). The *Android* sample contained $12 \cdot 40 = 480$ links, because we added four additional documentation resources that were only exhibited in that domain (see Table 3). Section 8 discusses implications of this sampling approach.

5 Qualitative Analysis

We qualitatively analyzed all links in our samples to build a first layer of interpretation for linking practices. Following our research questions, we organized the coding [34] along two dimensions, *context* and *purpose*. For analyzing the context, much information is already available directly in the posts (e.g., the text surrounding the links). For context, we designed the coding task to complement this information with insights that are impossible to extract automatically, namely, whether the text in the context includes a quote or a summary of the link target—or whether the link is provided without any context. For purpose, we were interested in producing an abstraction of the purpose of the link as it would appear to a third party who read the corresponding thread.

2. https://pages.github.com/

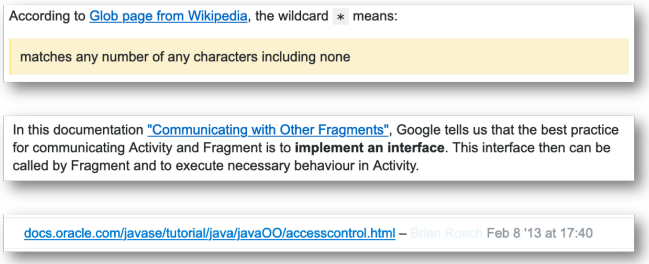

Fig. 2. Examples for context codes QUOTE (top), SUMMARY (middle), and LINKONLY (bottom), taken from Stack Overflow questions 37724969, 28949786, and comment 20689962.

Development of the Coding Guide

We developed a coding guide by considering the context and purpose dimensions separately.

Context

For the context, creating the coding guide amounted to agreeing on what constituted a quote, a summary, and a link without context. The task was thus to indicate, for each link in the sample, true or false as values for the attributes QUOTE and SUMMARY. The attribute QUOTE indicates the presence of non-trivial content that has been copied without modification from the linked documentation resource into the Stack Overflow post or comment, the attribute SUMMARY indicates that the Stack Overflow author provided at least one key insight from the linked documentation resource in their own words. The third context code LINKONLY was assigned in case only the URL was provided (including anchor text) without any additional information surrounding it. Note that while the codes QUOTE and SUMMARY can be assigned independent of the purpose codes, LINKONLY makes deriving a purpose impossible because no context is provided. Therefore we modeled the former two as independent binary codes, as outlined above (see also Tables 6 and 7). Figure 2 illustrates the difference between the three context codes we assigned.

Purpose

The development of a reliable coding guide for a link's purpose was much more challenging, and required multiple iterations. In an *initial coding* phase, we built a coding guide using a subset of the links for *Regex*. During the initial coding, all three authors coded 80 links in four tasks of 20 each, discussing emerging categories after completing each task, until a stable coding guide emerged. Prior to starting with the *Android* sample, all three authors coded 50 links and then discussed if changes to the coding guide were required, which only led to one minor addition. Note that, while the codes are not mutually exclusive, the coders always assigned one code that they considered to most accurately describe the link purpose. Table 4 lists the codes with a brief description. The full description can be found in the supplementary material. The modification that was required for the *Android* case was simply to add "watching a video" to the code BACKGROUNDREADING, because of the new documentation resource *YouTube*.

TABLE 4
Code catalog for link context and purpose (summary).

Abbrev.	Code	Description (Excerpt)
QUO	QUOTE	Presence of non-trivial content copied without further modification.
SUM	SUMMARY	At least one key insight from the linked resource is provided in paraphrased from.
ONL	LINKONLY	Link that only contains the URL (and anchor text) without any additional information surrounding it.
ATT	ATTRIBUTION	Link to a resource simply to credit the source for material taken verbatim.
AWA	AWARENESS	Link intended to make readers aware that a certain resources exists, or provide information about the nature of its content, without necessarily endorsing it.
BGR	BACKGROUNDREADING	Link to a resource that a user thinks other users should read or watch to get better general knowledge of the topic related to the thread.
CPT	CONCEPT	Link to a resource that contains a general description of a concept that the reader should know about.
CST	CONSULTED	Link to documentation to indicate that it was consulted prior to posting.
LMN	LINKEDMENTION	Link to the element-level (class, method, field) Javadocs of an API element that is mentioned as part of the text, without more specific indication for the purpose of the link.
RCM	RECOMMENDATION	Link to resources that are landing pages for tools, libraries, API elements, or algorithms, for the purpose of recommending these.
REF	REFERENCE	Links to a resource to indicate the source of knowledge for an explicit claim, statement, or information conveyed in the post.
OTH	OTHER	Link whose purpose is other than can be captured by other codes, unclear, or unknown.

Coding Process

We used the coding guide in a *focused coding* phase to go over all links in the sample and code them according to the guide, which we provide as supplementary material. All three authors used the coding guide to independently code the links by opening the Stack Overflow thread in a web browser, locating the link, and analyzing the surrounding context.

We coded the links in sets of up to 100 links, computing inter-rater agreement and discussing results after each set to ensure there were no major divergences or misunderstandings of the coding guide. To measure our inter-rater agreement, we calculated a three-way Cohen's kappa (κ) [35] for each set. Table 5 presents the agreement data.

TABLE 5
Inter-rater agreement for link purpose coding, with number of items in the set (#) and corresponding κ value.

	Regex			*Android*					
#	100	100	79	50	100	100	100	100	30
κ	0.61	0.65	0.77	0.71	0.70	0.64	0.74	0.72	0.80

The task of identifying the *purpose* of a link turns out to be very challenging. In some cases, the purpose can be ambiguous or opaque. The difficulty of the task is reflected in the kappa values. Although they increase towards the end as we became more proficient, values in the 0.65-0.80 range, although usable, are indicative of a non-negligible amount of residual flexibility of interpretation.

The difficulty of the coding task is the reason we opted for the unusual and very labor intensive practice of coding every single item in our data set in triplicate. This decision significantly mitigates the threats of bias in the coding task, since we were able to systematically detect links with ambiguous purpose and resolve disagreements by applying the following formal process: After each coding iteration, we merged the purpose and LINKONLY codes by selecting the code which at least two investigators used (majority vote), and assigned the code OTHER if there was no agreement, which happened for 14 *Regex* links (5%) and for 13 *Android* links (2.7%). The binary codes capturing the link *context* were assigned a value of true if at least two investigators considered the link to be accompanied by a QUOTE or SUMMARY respectively.

Final Coding

Tables 6 and 7 show the frequency of each code per documentation resource for both cases. Examples for the three context codes can be found in Figure 2. Section 6 presents examples for the purpose codes together with related developer resources.

While our URL mapper was able to detect most invalid or dead links, we still noticed some broken links in the samples (coded as N/A). We also coded links as N/A if they were not rendered on Stack Overflow's website, but present in the Markdown source of the posts or comments, which we used to extract the links from.

6 QUANTITATIVE ANALYSIS

The qualitative analysis provides the foundation that enabled three quantitative analyses to better understand linking practices:

1) A systematic comparison of code distributions between our two cases, to relate differences to their context.
2) The mining of *association rules* to detect correspondences between a resource type and a link purpose.
3) The building of logistic regression models, using question features as independent variables and presence of a link to a particular resource as dependent variable, to determine the characteristics of a Stack Overflow question that are related to the features of documentation links in an answer or a comment.

TABLE 6
Documentation resources and corresponding codes (purpose and context) for *Regex* case.

	ATT	AWA	BGR	CPT	CST	RCM	REF	LMN	ONL	OTH	N/A	Total	QUOTE	SUMMARY
StackOverflow	2	16	0	0	5	0	3	0	2	12	0	40	3	6
JavaAPI	2	4	0	0	5	5	7	12	1	3	1	40	7	10
IndependentTut.	1	2	9	9	6	0	8	0	0	4	1	40	3	5
JavaReference	2	4	12	3	6	2	6	1	1	3	0	40	3	7
Wikipedia	1	1	2	22	1	3	3	0	1	4	2	40	1	9
OtherAPI	1	6	0	0	3	17	0	8	0	4	1	40	2	6
OtherReference	1	13	2	3	2	3	3	0	0	7	2	36	2	7
OtherForum	0	1	0	0	0	0	0	0	0	1	1	3	0	0
Total	10	47	25	37	28	30	30	21	5	38	8	**279**	21	50

TABLE 7
Documentation resources and corresponding codes (purpose and context) for *Android* case.

	ATT	AWA	BGR	CPT	CST	RCM	REF	LMN	ONL	OTH	N/A	Total	QUOTE	SUMMARY
StackOverflow	1	23	0	0	4	0	3	0	2	6	1	40	0	3
JavaAPI	3	4	0	1	3	14	3	11	0	1	0	40	3	7
IndependentTut.	0	16	4	2	10	1	1	0	3	2	1	40	0	1
JavaReference	3	6	15	3	5	0	4	1	2	1	0	40	5	5
Wikipedia	1	7	2	22	0	1	1	0	0	6	0	40	2	4
OtherAPI	0	7	1	0	5	11	5	7	0	4	0	40	2	3
OtherReference	1	12	0	2	6	4	7	0	2	4	2	40	2	8
OtherForum	1	22	0	0	9	0	2	0	0	6	0	40	1	3
AndroidAPI	2	8	2	0	3	10	8	7	0	0	0	40	4	5
AndroidReference	3	11	11	0	4	4	2	1	1	2	1	40	3	5
AndroidIssue	2	20	0	0	3	0	6	0	0	9	0	40	1	6
YouTube	0	13	9	0	9	0	0	0	2	7	0	40	0	2
Total	17	149	44	30	61	45	42	27	12	48	5	**480**	15	34

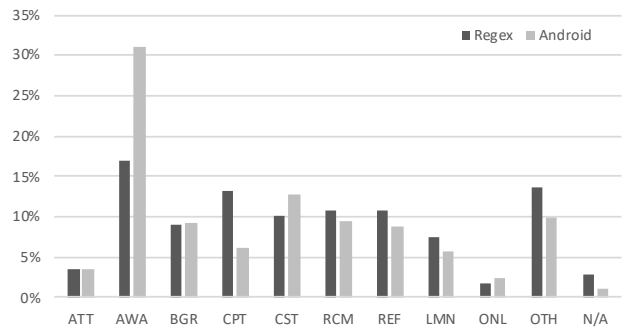

Fig. 3. Relative frequency of the assigned link purpose codes for both cases ($n_{\text{regex}} = 279,\ n_{\text{android}} = 480$).

Code Frequency Comparison

Figure 3 shows the relative frequency of the purpose codes we assigned.[3] The bar charts reveals two major differences: in our sample, the code AWARENESS was about twice as common in the *Android* case than in the *Regex* case (31.0% vs. 16.8%). The reverse was true for the code CONCEPT, which was about twice as common in the *Regex* case (13.3% vs. 6.3%). Both difference were significant according to a two-tailed Fisher's exact test [36] with a significance level of $\alpha = 0.01$.[4]

Both of these differences can be directly linked to salient aspects of the technological environment of the cases analyzed. The sample for the *Regex* case exhibits twice as many CONCEPT-related links, which can be explained by the theoretical nature of the domain. The links we coded are to concepts such as *context-free grammar* and *regular language*. As for *Android*, the extensive use of links for AWARENESS purposes can be explained by the huge size of this technology ecosystem, where many users end up posting answers and comments simply to point out relevant resources to each other.

Association Rule Mining

To distill the main motivation behind linking to documentation resources of a certain type, we mined association rules between resources and assigned purpose codes.

Method

We first transformed the documentation resource categories as well as the purpose and LINKONLY codes into binary properties of the links, added the QUOTE and SUMMARY codes, and then applied the *apriori* algorithm [37] as implemented in the R package arules[5] to retrieve binary rules.

3. Our use of stratified random sampling precludes the calculation of confidence intervals, which rely on an assumption of simple random sampling. As stated in the main text, the figure thus documents the code we assigned, without the implication that they would generalize to a population. This is consistent with our research goal and use of a case study method, whereby we sought to understand the phenomenon of link sharing as broadly as possible for two specific topics, as opposed to drawing implications for an entire dataset.

4. The p-values were 0.0001 for the AWARENESS frequency difference and 0.0014 for the CONCEPT frequency difference.

5. https://cran.r-project.org/web/packages/arules/index.html

TABLE 8
Binary association rules between documentation resource type and purpose/context codes in the *Regex* sample.

LHS	RHS	Supp	Conf	Lift	n
Wikipedia	→ CONCEPT	0.08	0.58	4.22	22
OtherAPI	→ RECOMM.	0.06	0.45	3.92	17
StackOverflow	→ AWARENESS	0.05	0.40	2.30	16
OtherReference	→ AWARENESS	0.05	0.38	2.20	13
JavaAPI	→ LINKEDMENTION	0.04	0.31	3.96	12
JavaReference	→ BACKGROUNDR.	0.04	0.30	3.24	12
Attribution	→ QUOTE	0.04	1.00	12.85	10
Reference	→ SUMMARY	0.08	0.73	3.96	22

TABLE 9
Binary association rules between documentation resource type and purpose/context in the *Android* sample; rules only present in this sample are highlighted with a gray background.

LHS	RHS	Supp	Conf	Lift	n
StackOverflow	→ AWARENESS	0.05	0.59	1.9	23
Wikipedia	→ CONCEPT	0.05	0.56	8.7	22
OtherForum	→ AWARENESS	0.05	0.55	1.8	22
AndroidIssue	→ AWARENESS	0.04	0.50	1.6	20
IndependentTut.	→ AWARENESS	0.03	0.41	1.3	16
JavaReference	→ BACKGROUNDR.	0.03	0.38	4.0	15
JavaAPI	→ RECOMM.	0.03	0.35	3.7	14
Youtube	→ AWARENESS	0.03	0.33	1.0	13
OtherReference	→ AWARENESS	0.03	0.32	1.0	12
AndroidReference	→ BACKGROUNDR.	0.02	0.28	3.0	11
AndroidReference	→ AWARENESS	0.02	0.28	0.9	11
JavaAPI	→ LINKEDMENTION	0.02	0.28	4.8	11
OtherAPI	→ RECOMM.	0.02	0.28	2.9	11
IndependentTut.	→ CONSULTED	0.02	0.26	2.0	10
AndroidAPI	→ RECOMM.	0.02	0.25	2.6	10
Attribution	→ QUOTE	0.03	0.88	18.22	15
Reference	→ SUMMARY	0.07	0.74	6.74	31

We note that the maximum support of a mined association rule is limited by the fact that we only sampled up to 40 links per documentation resource. The *Regex* sample, for example, contained 279 links in total (see Table 8). If a rule is true for all 40 links to one particular resource, the support would still only be $^{40}/_{279} = 0.14$. In our analysis, we considered rules with at least 10% of the maximum possible support, which was $^{0.14}/_{10} = 0.014$ for the regex sample and $^{0.08}/_{10} = 0.008$ for the *Android* sample. Moreover, we excluded rules with less than 25% confidence, meaning that a rule must be true in at least 1 out of 4 cases, and we further excluded rules involving the code OTHER.

Results

Tables 8 and 9 show the binary association rules between the documentation resource types and the purpose/context codes. In the following, we discuss those rules and provide illustrating examples.

Wikipedia → CONCEPT

The purpose CONCEPT was clearly associated with the resource *Wikipedia*, having the highest and second highest confidence in the two samples, respectively. A typical usage scenario was to mention a concept related to the question and then use the first mention of the concept as link anchor pointing to the corresponding page on Wikipedia:

> I think you're using * as if it's the **Kleene star**, not * as Java, JavaScript, & co. interpret * in regexps. [38]

This observation provides a clear characterization of the extent to which Wikipedia is leveraged to avoid defining concepts. The observation directly corroborates that of Vincent et al. [9], who found that "on SO, Wikipedia supports answers in the form of links and quoted text. Answers often use technical terms or acronyms and include a Wikipedia link in lieu of defining these terms."

Java-/OtherAPI → RECOMMENDATION

A second dominant group of association rules are related to RECOMMENDATIONS, which often pointed directly to the API documentation of a recommended software component. This is represented by the rule *OtherAPI* → RECOMMENDATION in the regex sample and *JavaAPI*/*OtherAPI* → RECOMMENDATION in the *Android* sample.

> You could use **Apache Commons Lang** for that... [39]

Java-/AndoidReference → BACKGROUNDREADING

A main use case of reference documentation was providing readers with pointers to resources for BACKGROUNDREADING. This relationship is also reproduced in the association rules we identified, since *JavaReference* were associated with BACKGROUNDREADING in both samples. Moreover, *AndroidReference* was associated with this purpose in the second sample. An example for BACKGROUNDREADING is provided below:

> Instead of asking people to code your regular expressions for you, try reading the Java Regular Expressions Tutorial. **...docs.oracle.com/javase/tutorial/...** [40]

The above example illustrates the difference between the codes RECOMMENDATION and BACKGROUNDREADING. We used RECOMMENDATION to highlight that the authors' primary intention was to recommend a specific tool or library (like in the example). BACKGROUNDREADING, on the other hand, indicates that the author recommends a certain resource describing background knowledge relevant for the topic of the particular thread (see also descriptions of the codes in Table 4).

StackOverflow → AWARENESS

Other rules for link purposes were not as insightful because they rather confirmed the definition of our codes than indicated a particular linking practice. For example, although *StackOverflow* → AWARENESS was a strong rule for both cases, it is hardly surprising that people will link to a Stack Overflow post to make others aware of it.

QUOTE/SUMMARY/LINKONLY

Regarding the *context* of links, we only identified two rules that were present in both samples: *Attribution* → QUOTE and *Reference* → SUMMARY. The former indicates an obvious relationship between content copied from external sources and the purpose of attributing that content. The latter indicates that especially for reference documentation, Stack Overflow authors felt the need to summarize key insights instead of copying content as-is.

Overall, quoting content was not very common in the posts and comments we analyzed. In the *Regex* sample, 7.5% of the links referred to content being quoted, in the *Android* sample only 3.1% (see Table 6). The quoted content ranged from complete code snippets to small parts of the reference documentation. Summarizing linked resources was more common than quoting (17.9% in *Regex* and 7.1% in *Android*). However, there was neither a summary nor a quote for 203 *Regex* (72.8%) and 400 *Android* links (83.3%), which can become a problem once the links are dead.

Model Building

To investigate which properties of a Stack Overflow question might explain whether it will attract documentation links, we built separate logistic regression models for the *Regex* and *Android* cases.

Data Preparation

For each of the two cases (*Regex* and *Android*), the input data for the model building were three samples, each containing 100 Stack Overflow threads:

- ***Documentation links:*** One sample with threads that attracted links to documentation resources. To identify such threads, we relied on our previous classification and randomly selected 100 threads with at least one answer or comment containing a link classified as pointing to one of the documentation resources (see Table 3).
- ***Non-documentation links:*** One sample with threads that attracted links, but not to documentation resources. We randomly selected 100 threads with at least one answer or comment containing a non-classified or non-documentation link (see Table 3).
- ***No links:*** One sample with threads that did not attract links at all. To draw this sample, we utilized the *SOTorrent* dataset and selected only threads without any links in answers and comments (no records in tables PostVersionUrl and CommentUrl).

Our data retrieval and sampling scripts are available as part of the supplementary material. Two of the authors independently analyzed all 600 threads to verify that they are indeed a representative of the corresponding class. In case we found contradicting evidence (e.g., a link to a documentation resource in one of the non-documentation samples), we excluded those threads and then sampled and analyzed replacements.

Non-documentation Resources

In the course of analyzing the two non-documentation samples, we also coded the purposes of those links. In the *Regex* sample, the most common purposes of non-documentation links were referring to a (regex) *tool* (46), *source code* (19), or websites with *posting recommendations*[6] (16). In the *Android* sample, the most common purposes were linking *source code* (28), an online *tool* (22, e.g., JSON or XML validators), or an *image file* (19, e.g., icons or screenshots).

6. Examples: http://whathaveyoutried.com/ or http://sscce.org/

TABLE 10
Features of Stack Overflow posts used as independent variables in the logistic regression models.

Feature	Description
TitleLength	# of characters in question title
TextBlockCount	# of text blocks in question
CodeBlockCount	# of code blocks in question
LineCountText	# of lines of text in question
LineCountCode	# of lines of code in question
LengthText	# of characters formatted as text
LengthCode	# of characters formatted as code
UserAgeWhenPosting	# of days since account creation
UserReputation	reputation of user
LinkCount	# of links in question
LinkSpecificity	0: no link 1: link to root domain 2: path present 3: path contains fragment identifier
Tags *(one feature per tag)*	tags associated with the question Regex*: 4 features,* Android*: 3 features*
Words in title *(one feature per word)*	the question title Regex*: 14 features,* Android*: 2 features*
Words in body *(one feature per word)*	all text in the question body Regex*: 86 features,* Android*: 69 features*
Terms in code *(one feature per term)*	all code in the question body Regex*: 23 features,* Android*: 118 features*

Features

Table 10 shows the features used as independent variables in the logistic regression models. The set of features consists of numeric features that can be extracted from the question, such as *LengthText* or *CodeBlockCount*. Note that we excluded features that would be unknown at the time when the question was posted, such as how many views the question attracted or its score. We retrieved the data for the features from the *SOTorrent* dataset, which contains the content of Stack Overflow posts separated into text and code blocks, collects links from posts and questions, and provides the metadata from the official Stack Overflow data dump.

For the textual features, shown in the bottom part of Table 10, we treated each token as a separate feature and used token frequency as feature values. We separated text into tokens using whitespace, and we removed stopwords[7] and punctuation as well as special characters. All tokens were stemmed using the Porter stemming algorithm [41]. We discarded features consisting of a single character such as a single digit, and we limited the set of features to tokens whose frequency in our dataset exceeded a minimum threshold. We used the goodness of fit (measured using McFadden's pseudo-R^2 [42]) to determine the best threshold for each dataset, resulting in a threshold of 15 for the *Regex* dataset (McFadden's pseudo-R^2 = 0.549) and 22 for the *Android* dataset (McFadden's pseudo-R^2 = 0.592). This led to a total of 138 features for the *Regex* dataset and 203 features for the *Android* dataset. Table 10 shows the number of features resulting from each textual property.

The interpretation of logistic regression models may be misleading if the metrics that are used to construct them are correlated [43]. As Table 10 shows, some of our features are likely to be correlated, e.g., *LineCountText* and *LengthText*. To

7. We used the "Long Stopword List" from https://www.ranks.nl/stopwords

TABLE 11
Most important features for explaining whether a Stack Overflow question related to regular expressions will attract a particular type of documentation link. All p-values < 0.001.

Resource	Origin	Feature	Coeff.	ANOVA
Wikipedia	Text	pars	+13.1	9%
	Text	java	+10.2	8%
	Text	issu	+21.9	7%
	Title	pattern	+15.4	6%
	Text	problem	−16.9	5%

TABLE 12
Most important features for explaining whether a Stack Overflow question related to *Android* will attract a particular type of documentation link. All p-values < 0.001.

Resource	Origin	Feature	Coeff.	ANOVA
Wikipedia	Text	devic	+15.7	17%
	Text	creat	+7.0	11%
	Text	user	+9.0	5%
	Metadata	UserReputation	+0.0	4%
	Text	call	−11.4	3%
Stack Overflow	Text	find	+46.8	8%
	Code	activ	−7.2	7%
	Text	click	−45.6	6%
	Text	call	−17.6	6%
	Code	edittext	+19.9	6%
JavaAPI	Text	convert	+42.3	10%
	Text	phone	−76.8	6%
	Text	problem	+38.3	6%
	Text	string	+8.7	6%
	Metadata	LineCountText	+4.4	5%
Android *Reference*	Metadata	UserReputation	+0.0	11%
	Code	text	−29.7	6%
	Code	065941702	+6.8	5%
	Code	wsystemerr1249	+7.1	5%
	Code	viewonclicklisten	+58.7	4%

mitigate correlated metrics, we used AutoSpearman [44], an automated metric selection approach based on correlation analyses, with a threshold of 0.7.

Following the advice of Tantithamthavorn and Hassan [43], we used ANOVA Type-II importance scores to interpret our logistic regression models after constructing them using the `glm` function in R.

Models For Documentation Resources

We built logistic regression models for specific types of documentation resources. Note that we do not treat type of documentation resource as a categorical variable since posts can contain links to multiple different documentation resources. While we did not have enough data to allow the construction of models for all types of resources, Tables 11 and 12 show the five most important features (as determined by the ANOVA Type-II test) for a subset of resource types for the *Regex* and *Android* datasets. Table 11 indicates that *Regex* questions about *parsing* and *patterns* are associated with a higher chance of attracting a link to *Wikipedia*. In contrast, questions about specific *problems* are associated with a lower likelihood. For *Android*, questions about *devices* are associated with a higher chance of attracting *Wikipedia* links while questions about *converting* are associated with attracting links to the *JavaAPI*. As shown in Table 12, links

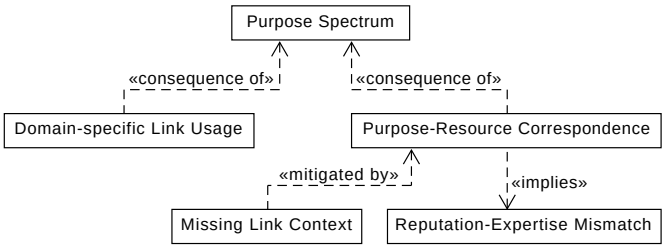

Fig. 4. Relationships between findings about linking practices on Stack Overflow.

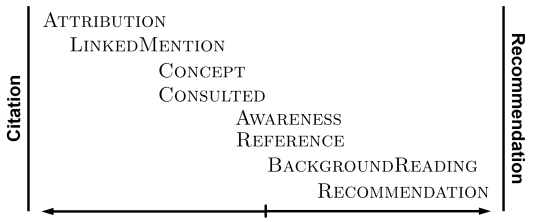

Fig. 5. Purpose codes arranged on the purpose spectrum from citation to recommendation.

to the AndroidReference documentation are associated with questions asked by users with a higher reputation. Interestingly, a manual inspection of the corresponding questions suggests that many of these high-reputation users are outsiders whose expertise is in areas other than *Android*.

We provide an interpretation of these results as part of the discussion in the next section. Ultimately, such models could be used to recommend the inclusion of different types of links in Stack Overflow posts.

7 Findings

Our systematic analysis of the *context* (**RQ1**) and *purpose* (**RQ2**) of documentation links led to five major findings about linking practices on Stack Overflow. In this section, we detail the evidence for each finding and discuss its main implication.

Purpose Spectrum

Our qualitative analysis has shown that documentation links on Stack Overflow serve a variety of purposes. Figure 3 shows a rich diversity of purposes with eight of ten categories showing relative frequency above 5%. Manually reviewing all the links (through the coding process) also showed that the different categories of link purposes can be positioned on a spectrum bounded by the concepts of *Citation* and *Recommendation*, where citations are not meant to be consulted whereas recommendations are explicit entreaties to follow the link. Figure 5 positions every link purpose category except for Other along this axis.

Citation links include the ones labeled as Attribution and LinkedMention. The purpose of Attribution links is to credit the source of content copied into Stack Overflow, which can help users meet Stack Overflow's requirement for attribution [45]. The purpose of the LinkedMention links is to uniquely identify a software artifact without the need

to provide further context. Often, users add such LINKED-MENTION references as inline links, which underlines their peripheral role:

> Is there a regex that would work with **String.split()** to break a String into contiguous characters...? [46]

We place CONSULTED and CONCEPT in the middle of the spectrum because they are open to interpretation. CONSULTED links are typically added for context, but in some cases this context is simply to show due diligence (closer to citation) and in some cases it is to point to an unclear document to be explained, e.g.,:

> I am trying to understand the regular expression in Solr and came across **this** Java doc where explains... having a hard time understanding what it really means. [47]

As for CONCEPT links, they are useful for readers who want to learn more about a mentioned concept, but they are usually also peripheral to the actual content of the post or comment (reproduced from Section 6).

> I think you're using * as if it's the **Kleene star**, not * as Java, JavaScript, & co. interpret * in regexps. [38]

Closer towards *Recommendation* we place AWARENESS links that steer users' attention towards related resources, without particularly endorsing them, as well as REFERENCE links that users include to make statements verifiable and more trustworthy by pointing to documentation resources supporting their claims.

One purpose of links towards the *Recommendation* end of the spectrum is to explicitly guide readers to BACKGROUNDREADING. Such links are especially helpful for users who are new to a topic or domain since they support them in identifying relevant background knowledge:

> There is a good detailed description of lookarounds (look-behind and look-ahead) as well as a lot of other regex "magic" **here** [48]

Finally, we find explicit RECOMMENDATION links. They allow readers to retrieve a specific software component recommended by a Stack Overflow author using the provided link (reproduced from Section 6).

> You could use **Apache Commons Lang** for that... [39]

Implication: Forum users add links to documentation for a variety of purposes. This purpose may not be clear to the reader. Links whose purpose is not clear may confuse or waste the time of inexperienced users, who are surmised to visit more links as they navigate web sites [49]. Automated analysis of link data (e.g., [8]) may miss opportunities for additional interpretation if link purpose is not taken into account.

Purpose–Resource Correspondence

In the two cases we studied, mined association rules show consistent relations between a resource type (e.g., Wikipedia, Stack Overflow) and a link's purpose. Links to *Wikipedia*, for example, often serve to define CONCEPTs, an observation consistent with previous work [9]. Links to the documentation of software components and tools are often included to recommend the tool rather than to refer to the linked document specifically (RECOMMENDATION).

Implication: For technology domains where certain resource types can be strongly associated with a link purpose, it may be possible to automatically recommend links to enhance a post, or infer the purpose of a linked resource.

Domain-specific Link Usage

The distribution of link purposes shown in Figure 3 and detailed in Tables 6–7 shows remarkable consistency between cases except for two major differences: Purpose code CONCEPT is about twice as common in case *Regex* and purpose code AWARENESS is about twice as common in case *Android*. For the other codes, the relative frequency differs not more than 3 percentage points. Both differences mentioned above are significant at the level $\alpha = 0.01$ (see Section 6). From this we conjectured that the higher proportion of CONCEPT links is explained by the theoretical nature of the domain, which involves concepts such as "parsing", "context-free grammar", "pattern", etc. This observation is corroborated by the regression model, which shows that one of the dominant features for explaining whether a Stack Overflow question related to regular expressions will attract a particular type of documentation link include such theoretical concepts, namely "parsing" and "pattern". As for Android, the extensive use of links for AWARENESS purposes can be explained by the size of this technology ecosystem.

As mentioned above, we added the documentation resource *Youtube* while adapting the classifier for the *Android* case. This is another manifestation of domain-specific link usage, because in the *Regex* case, only 26 posts pointed to *Youtube* (0.09% of all posts containing links), while in the *Android* case, linking *Youtube* videos was much more common (1,822 posts or 0.8% of all posts containing links). The difference was significant according to a two-tailed Fisher's exact test [36] with a significance level of $\alpha = 0.01$ (p-value $< 2.2 \times 10^{-16}$). Typical use cases of linking *Youtube* videos include pointing to tutorials[8] or conference talks.[9]

Implication: Links to documentation resources are a reflection of the information needs typical to a technology domain. Details on the distribution of purpose links for a domain can thus assist in the design of documentation.

Missing Link Context

Even though Stack Overflow encourages users to provide context for links [10], they are rarely accompanied by a QUOTE [28] or a SUMMARY. Our analysis shows that, for 72.8% of the analyzed links, authors did not provide a quote and for 83.3% of the links they did not provide a summary. Although in some situations this lack of context may render links worthless once their target is unavailable, our analysis also revealed valid use cases for links without context, as links at the *Citation* end of the purpose spectrum

8. Example tutorial: https://youtu.be/fn5OlqQuOCk
9. Example conference talk: https://youtu.be/N6YdwzAvwOA

do not necessarily need context. However, links towards the *Recommendation* end of the spectrum should always be accompanied by additional information to preserve that information in case the linked resources becomes unavailable.

Implication: Our link *Purpose Spectrum* observation allows us to modulate the requirement to add context for links, given that our data shows the context to be self-explanatory for links whose purpose is akin to a citation. We hypothesize that the importance of context for orienting users is proportional to a link's position on the purpose spectrum. Missing context is thus not necessarily a problem for links whose purpose is citation.

Reputation-Expertise Mismatch

The logistic regression analysis shows that users with a high reputation score are not necessarily more familiar with reference documentation than lower reputation users. Links to the ANDROIDREFERENCE documentation are associated with questions asked by users with a higher reputation. The median user reputation of users asking questions which attract links to the ANDROIDREFERENCE documentation in the dataset used for the logistic regression analysis is 1063.5, while the corresponding median for the remaining questions is 86. A manual inspection of the corresponding questions suggests that many of these high-reputation users are outsiders whose expertise is, based on the questions they typically answer, in areas other than Android (often iOS). Similarly, links to WIKIPEDIA are also associated with questions asked by users with a higher reputation.

Implication: In previous research efforts, researchers have often treated an individual's reputation on Stack Overflow as a proxy for this individual's general programming knowledge (e.g., [50]). Our results indicate that this operationalization may not be valid in all scenarios, because Stack Overflow authors' knowledge is domain-specific.

Summary

The findings described in the previous paragraphs build on each other to form a small conceptual framework defined in terms of logical implications. Figure 4 summarizes the findings and their relationships.

Our primary finding concerns the variety of linking purposes we elicited and the observation that linking purpose types span a spectrum that characterizes to what extent a link is intended to be followed (*Purpose Spectrum*).

We also collected evidence of a notable correspondence between a resource type (e.g., Wikipedia) and a link's purpose (*Purpose–Resource Correspondence*), and that link usage may be specific to a technology domain (*Domain-Specific Link Usage*). Both of these observations are consequences of *Purpose Spectrum* in the sense that it is the observed richness of linking purposes that enables the elicitation of specific linking practices.

A fourth observation is the extent to which links in Stack Overflow threads lack context, despite the presence of guidelines explicitly requesting such context (*Missing Link Context*). To a certain extent, this observed problem can be mitigated by *Purpose–Resource Correspondence* because this correspondence supports partial inference of a link's purpose.

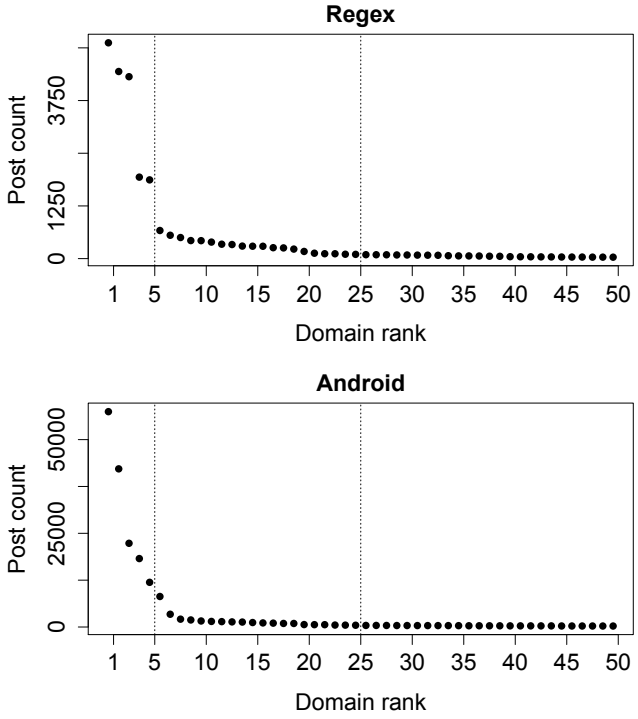

Fig. 6. Post count for top 100 root domains, top 5 shown in Tables 1 and 2, top 25 incorporated in classifier.

Finally, our analysis reveals a pattern that would be counter-intuitive at first glance: users with high reputation attract answers with links to the reference documentation, which can also be construed a symptom of lack of expertise (*Reputation-Expertise Mismatch*). This finding is enabled by the *Purpose–Resource Correspondence* which relates links to documentation resources with a type of information need.

8 THREATS TO VALIDITY

The external validity of our results may be limited due to our choice of the two specific domains *Regex* and *Android*, both of which were taken from the Java domain. While Java is one of the most popular programming languages today,[10] its documentation ecosystem may differ from other languages. The documentation resources we identified, such as API documentation [51] and Wikipedia [9], are, however, likely to also play an important role for other languages and domains.

Another threat is that our URL mapper was only able to classify 78.5% of all active links in the *Regex* sample and 68.9% of all active links in the *Android* sample (see Section 4). Note that a classification of the remaining links would only add more documentation resources, but not invalidate the ones we have already identified. Also, the number of posts containing a link to the corresponding

10. TIOBE Index for December 2019, https://www.tiobe.com/tiobe-index/, verified 16 December 2019.

root domain considerably drops after the top five (*Regex*) respectively the top six (*Android*) root domains (see Figure 6). This also means that the marginal profit of analyzing additional root domains drops considerably after analyzing the most frequently referenced root domains. While iteratively building the classifier, two authors continuously discussed the emerging documentation resource categories and corresponding sub-pages of the root domains. Since we followed a whitelisting approach based on regular expressions matching certain sub-paths of the domains, making all decisions unanimously, the false positive rate of our approach is very low. We may, however, have missed certain sub-paths, marking them as *NotDocumentation* when they were in fact documentation resources (false negatives). We mitigated this bias by manually inspecting the links marked as *NotDocumentation* after each iteration, filtering out links that clearly did not point to documentation resources until no links were left to analyze.

The stratified sampling strategy we used to select documentation links for our analyses represents a threat to the external validity of our results. Note that in a random sample, the top three documentation resources, especially the internal Stack Overflow links, would overshadow the less frequent documentation resources (see Table 3). Our sampling strategy allowed us to analyze a broader and more diverse sample of documentation resources not dominated by those very frequent link targets. In the association rule analysis we conducted, support and confidence only hold for our samples—they would differ in non-stratified samples. In Section 6, we described how to interpret those values considering the stratification. Moreover, the fact that all rules derived from the *Regex* sample were also present in the *Android* sample further supports their credibility.

The purpose distribution would likely differ in a random sample. However, in a random sample, frequently referenced documentation resources such as *Stack Overflow*, *JavaAPI*, and *AndroidAPI* would dominate the analysis. The stratification allowed us to consider a more diverse range of resources and purposes.

Qualitative data analysis always depends on the imagination and perception of the researcher. To mitigate this threat, all three authors conducted the qualitative analysis independently. We coded links in sets of up to 100 links, thoroughly discussed our results after finishing each set, assessed the inter-rater agreement, and only assigned a code if at least two researchers agreed on it.

9 Conclusion

Over the past decade, the community question answering platform Stack Overflow has become extremely popular among programmers for finding and sharing knowledge. However, the site does not exist in isolation, and users frequently link to other documentation sources, such as API documentation and encyclopedia articles, from within questions, answers, or comments on Stack Overflow. To understand how and why documentation is referenced from Stack Overflow threads, we conducted a multi-case study of links in two different technology domains, regular expressions and Android development. We used qualitative and quantitative research methods to systematically investigate the context and purpose of a sample of 759 documentation links.

We identified a spectrum of purposes for which links are included in Stack Overflow threads, ranging from ATTRIBUTION and LINKEDMENTION on the citation end of the spectrum to BACKGROUNDREADING and RECOMMENDATION of software artifacts on the recommendation side. Citations are not necessarily meant to be consulted whereas recommendations are explicit requests to follow a link. This observation relates to Stack Overflow's recommendation to add context to every link: While adding context in the form of summaries or quotes is important for links on the recommendation end of the purpose spectrum, it is less important for links primarily included for citation purposes.

We also found that links to documentation resources are a reflection of the information needs typical to a technology domain. For example, CONCEPT links were twice as common in threads about regular expressions compared to Android, while we found the opposite for AWARENESS links. These insights can inform the design and customization of documentation for different technology domains.

Our work forms a first step towards understanding how and why documentation resources are referenced on Stack Overflow, with the ultimate goal of improving the efficiency of information diffusion between Stack Overflow and the broader software documentation ecosystem, as motivated in Section 2. In the short term, Stack Overflow authors can use our results to reflect on the intended purpose before posting a link, and to learn how they can make their post more valuable by providing context.

Another direction for future work is developing tool support for guiding Stack Overflow users to enhance (potential) information diffusion. One tool could assist readers of Stack Overflow threads by automatically classifying links in posts or comments along the purpose spectrum we presented in this paper. Such a tool could be implemented as a browser plugin visualizing the determined purpose of the link, helping users to judge whether the link it is worth following based on their particular needs. Another idea is to extend the models we presented in Section 6 to be able to recommend Stack Overflow authors to include a certain type of link while creating or revising Stack Overflow posts.

References

[1] J. Spolsky, "Stack Overflow launches," Blog Post https://www.joelonsoftware.com/2008/09/15/stack-overflow-launches/, accessed: 21 August 2018.

[2] L. Mamykina, B. Manoim, M. Mittal, G. Hripcsak, and B. Hartmann, "Design lessons from the fastest Q&A site in the west," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2011, pp. 2857–2866.

[3] C. Treude, O. Barzilay, and M.-A. D. Storey, "How do programmers ask and answer questions on the web?" in *33rd International Conference on Software Engineering (ICSE 2011)*, R. N. Taylor, H. C. Gall, and N. Medvidovic, Eds. Waikiki, Honolulu: ACM, 2011, pp. 804–807.

[4] C. Parnin, C. Treude, L. Grammel, and M.-A. Storey, "Crowd documentation: Exploring the coverage and the dynamics of api discussions on stack overflow," *Georgia Institute of Technology, Tech. Rep*, 2012.

[5] W. Maalej and M. P. Robillard, "Patterns of knowledge in API reference documentation," *IEEE Transactions on Software Engineering*, vol. 39, no. 9, pp. 1264–1282, 2013.

[6] M. P. Robillard and R. Deline, "A field study of API learning obstacles," *Empirical Software Engineering*, vol. 16, no. 6, pp. 703–732, 2011.

[7] M. Meng, S. Steinhardt, and A. Schubert, "How Developers use API Documentation: An Observation Study," in *Communication Design Quarterly Review*, vol. 7, 2019, pp. 40–49.

[8] C. Gómez, B. Cleary, and L. Singer, "A study of innovation diffusion through link sharing on Stack Overflow," in *Proceedings of the 10th IEEE Working Conference on Mining Software Repositories*, 2013, pp. 81–84.

[9] N. Vincent, I. Johnson, and B. Hecht, "Examining wikipedia with a broader lens: Quantifying the value of Wikipedia's relationships with other large-scale online communities," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 2018, pp. 566:1–566:13.

[10] "How do i write a good answer?" Stack Overflow Help Center https://stackoverflow.com/help/how-to-answer, accessed: 04 February 2019.

[11] Stack Overflow, "Searching for both word and its negation in a string using java regex," https://stackoverflow.com/q/21761788/.

[12] J. Li, Z. Xing, and A. Sun, "Linklive: discovering web learning resources for developers from q&a discussions," *World Wide Web*, pp. 1–27, 2018.

[13] J. Slegers, "The decline of Stack Overflow: How trolls have taken over your favorite programming q&a site," Hackernoon Blog Post https://hackernoon.com/the-decline-of-stack-overflow-7cb69faa575d, accessed: 21 August 2018.

[14] Stack Overflow, "Self-signed certificate on android," https://stackoverflow.com/q/24121224.

[15] C. Treude and M. P. Robillard, "Understanding Stack Overflow Code Fragments," in *2017 IEEE International Conference on Software Maintenance and Evolution (ICSME 2017)*, H. Mei, L. Zhang, and T. Zimmermann, Eds. Shanghai, China: IEEE Computer Society, 2017, pp. 509–513.

[16] Stack Overflow, "Mongo find() with regex in java only return one entry," https://stackoverflow.com/a/24890987.

[17] ——, "Java regular expression to discover regular expression," https://stackoverflow.com/q/30910685.

[18] S. Gao, Z. Xing, Y. Ma, D. Ye, and S.-W. Lin, "Enhancing Knowledge Sharing in Stack Overflow via Automatic External Web Resources Linking," in *2017 22nd International Conference on Engineering of Complex Computer Systems (ICECCS)*, Nov. 2017, pp. 90–99.

[19] B. Xu, D. Ye, Z. Xing, X. Xia, G. Chen, and S. Li, "Predicting Semantically Linkable Knowledge in Developer Online Forums via Convolutional Neural Network," in *Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering*, 2016, pp. 51–62. [Online]. Available: http://doi.acm.org/10.1145/2970276.2970357

[20] J. Yang, C. Hauff, A. Bozzon, and G.-J. Houben, "Asking the right question in collaborative Q&A systems," in *25th ACM Conference on Hypertext and Social Media (HT 2014)*, L. Ferres, G. Rossi, V. A. F. Almeida, and E. Herder, Eds. Santiago, Chile: ACM, 2014, pp. 179–189.

[21] L. Ponzanelli, A. Mocci, A. Bacchelli, and M. Lanza, "Understanding and classifying the quality of technical forum questions," in *14th International Conference on Quality Software (QSIC 2014)*, W. E. Wong and B. McMillin, Eds. Allen, TX, USA: IEEE, 2014, pp. 343–352.

[22] M. Duijn, A. Kucera, and A. Bacchelli, "Quality Questions Need Quality Code: Classifying Code Fragments on Stack Overflow," in *12th Working Conference on Mining Software Repositories (MSR 2015)*, M. Di Penta, M. Pinzger, and R. Robbes, Eds. Florence, Italy: IEEE Computer Society, 2015, pp. 410–413.

[23] F. Calefato, F. Lanubile, and N. Novielli, "How to ask for technical help? Evidence-based guidelines for writing questions on Stack Overflow," *Information and Software Technology*, vol. 94, pp. 186–207, 2018.

[24] S. Baltes, L. Dumani, C. Treude, and S. Diehl, "SOTorrent: Reconstructing and analyzing the evolution of Stack Overflow posts," in *Proceedings of the 15th International Conference on Mining Software Repositories (MSR 2018)*, 2018, pp. 319–330.

[25] D. Ye, Z. Xing, and N. Kapre, "The structure and dynamics of knowledge network in domain-specific Q&A sites: a case study of Stack Overflow," *Empirical Software Engineering*, vol. 22, no. 1, pp. 375–406, Feb. 2017.

[26] H. Hata, C. Treude, R. G. Kula, and T. Ishio, "9.6 million links in source code comments: Purpose, evolution, and decay," in *Proceedings of the 41st International Conference on Software Engineering*, 2019.

[27] J. Jiang, J. Cao, and L. Zhang, "An Empirical Study of Link Sharing in Review Comments," in *Software Engineering and Methodology for Emerging Domains*, ser. Communications in Computer and Information Science, Z. Li, H. Jiang, G. Li, M. Zhou, and M. Li, Eds. Springer, 2019, pp. 101–114.

[28] "How to reference material written by others," Stack Overflow Help Center https://stackoverflow.com/help/referencing, accessed: 04 February 2019.

[29] S. Baltes and L. Dumani, "SOTorrent dataset 2018-06-17," Jun. 2018. [Online]. Available: https://doi.org/10.5281/zenodo.1295405

[30] ——, "SOTorrent dataset 2018-07-31," Jul. 2018. [Online]. Available: https://doi.org/10.5281/zenodo.1401828

[31] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 1993, pp. 207–216.

[32] S. Baltes, C. Treude, and M. P. Robillard, "Contextual Documentation Referencing on Stack Overflow Supplementary Material," Feb. 2019. [Online]. Available: https://doi.org/10.5281/zenodo.2556642

[33] S. Baltes, M. P. Robillard, and C. Treude, "sbaltes/condor on GitHub," 2019. [Online]. Available: https://doi.org/10.5281/zenodo.2557446

[34] K. Charmaz, *Constructing grounded theory*, 2nd ed. Sage, 2014.

[35] J. Cohen, "A coefficient of agreement for nominal scales," *Educational and Psychological Measurement*, vol. 20, no. 1, pp. 37–46, 1960.

[36] R. A. Fisher, "On the interpretation of χ^2 from contingency tables, and the calculation of p," *Journal of the Royal Statistical Society*, vol. 85, no. 1, pp. 87–94, 1922.

[37] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules in large databases," in *Proceedings of 20th International Conference on Very Large Data Bases (VLDB 1994)*, 1994, pp. 487–499.

[38] Stack Overflow, "Finding the index of the first match of a regular expression in java," https://stackoverflow.com/q/8752252#comment10904903_8752252.

[39] ——, "how to check if string contains only numerics or letters properly? android," https://stackoverflow.com/a/34792055.

[40] ——, "Regular expression match a-alphanumeric&b-digits&c-digits," https://stackoverflow.com/q/17267166/.

[41] M. F. Porter, "An algorithm for suffix stripping," *Program*, vol. 14, no. 3, pp. 130–137, 1980.

[42] D. McFadden, "Conditional logit analysis of qualitative choice behavior," in *Frontiers in Econometrics*, P. Zarembka, Ed. New York, NY, USA: Wiley, 1973, ch. 4, pp. 105–142.

[43] C. Tantithamthavorn and A. E. Hassan, "An experience report on defect modelling in practice: Pitfalls and challenges," in *Proceedings of the 40th International Conference on Software Engineering: Software Engineering in Practice*. ACM, 2018, pp. 286–295.

[44] J. Jiarpakdee, C. Tantithamthavorn, and C. Treude, "Autospearman: Automatically mitigating correlated metrics for interpreting defect models," in *Proceedings of the 34th International Conference on Software Maintenance and Evolution*, 2018, to appear.

[45] S. Baltes and S. Diehl, "Usage and attribution of Stack Overflow code snippets in GitHub projects," *Empirical Software Engineering*, pp. 1–44, 2018.

[46] Stack Overflow, "Split regex to extract strings of contiguous characters," https://stackoverflow.com/q/13596454.

[47] ——, "Usage of — and :== in java doc," https://stackoverflow.com/q/35762611.

[48] ——, "Regex handling zero-length match," https://stackoverflow.com/a/28153330.

[49] A. Chevalier and M. Kicka, "Web designers and web users: Influence of the ergonomic quality of the web site on the information search," *International Journal of Human-Computer Studies*, vol. 64, no. 10, pp. 1031 – 1048, 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1071581906000838

[50] P. Morrison and E. Murphy-Hill, "Is programming knowledge related to age? An exploration of Stack Overflow," in *10th International Working Conference on Mining Software Repositories (MSR*

2013), T. Zimmermann, M. Di Penta, and S. Kim, Eds. San Francisco, CA, USA: IEEE, 2013, pp. 69–72.

[51] C. Parnin, C. Treude, and L. Grammel, "Crowd documentation: Exploring the coverage and the dynamics of api discussions on Stack Overflow," eorgia Institute of Technology, Tech. Rep., 2012.

Sebastian Baltes is a Lecturer in the School of Computer Science at the University of Adelaide, Australia. He received his PhD degree in Computer Science from the University of Trier, Germany. In his research, he empirically analyzes software developers' work habits to derive tool requirements or to identify possible process improvements. Most empirical studies he conducts combine qualitative and quantitative research methods and he is especially interested in interdisciplinary research, including the application of theories and methods from other disciplines in the software engineering domain.

Christoph Treude is an ARC DECRA Fellow and a Senior Lecturer in the School of Computer Science at the University of Adelaide, Australia. He completed his PhD in Computer Science at the University of Victoria, Canada, in 2012 and received his Diplom degree from the University of Siegen, Germany, in 2007. The goal of his research is to advance collaborative software engineering through empirical studies and the innovation of processes and tools that explicitly take the wide variety of artefacts available in a software repository into account.

Martin P. Robillard is a Professor of Computer Science at McGill University. His current research focuses on problems related to software evolution, architecture and design, and software reuse. He served as the Program Co-Chair for the 20th ACM SIGSOFT International Symposium on the Foundations of Software Engineering (FSE 2012) and the 39th ACM/IEEE International Conference on Software Engineering (ICSE 2017). He received his Ph.D. and M.Sc. in Computer Science from the University of British Columbia and a B.Eng. from École Polytechnique de Montréal.