Appendix of the Paper *"What do Developers Discuss about Code Comment Conventions?"*

Authors details omitted for double-blind reviewing

APPENDIX

A. Further details on mailing lists

TABLE I
KEYWORDS SEARCHED IN THE AGHAJANI *et al.* WORK [1]

Keyword	# Emails matched	Relevant mail threads	Relevant emails
comment	58	3	5
commenting	0	-	-
code comment	10	-	-
block comment	0	-	-
auto comment	0	-	-
comment convention	0	-	-
convention	0	-	-
naming	0	-	-
documentation	996	10% from random 100 sample	-
todo	0	-	-
code documentation	0	-	-
readability	0	-	-
code	29	-	-
variable	2	-	-
"document"	14	-	-
style	15	3	5

TABLE II
MAIL THREADS FOR KEYWORDS MATCHED TO THE AGHAJANI *et al.* WORK [1]

Keyword	Email subject	Discussion
comment	*code comment in TxnManagerImpl* [*river-dev/201304.mbox*]	Suggest improving comments
	Commented code in StartCatchAction [*ode-dev/200608.mbox*]	to know the intention behing a commented code
style	*javadoc style*[*commons-dev/201212.mbox*]	Asking for @see tag usage
	cvs commit: httpd-2.0/docs/manual/style modulesynopsis.dtd [*httpd-docs/200203.mbox*]	usage of "em" tag
	C++: API Documentation Style/-Tool [*parquet-dev/201604.mbox*]	Aksing for a tool for API documentation
documentation	*Documentations issues* [*orc-dev/201706.mbox*]	pointing out the isses found in comments about values of variables
	API documentation for SystemML [*systemml-dev/201512.mbox*]	Looking for a class documentation
	Very confusing documentation [*struts-user/201609.mbox*]	Different usage of English words in the comments

B. Detailed Taxonomy

Table III shows the hierarchy of categories constructed in the second dimension *i.e.*, *First level*, *Second level* with the definition (D) and one example (E) of each category in the column *Definition and example*.

C. Our study reproducibility

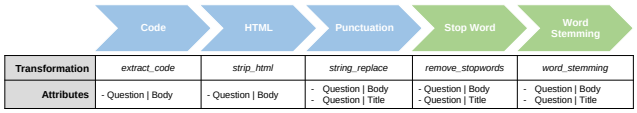
González *et al.* identified the reproducibility aspects characterizing empirical software engineering studies: Data Extraction,

TABLE III
TYPE OF INFORMATION DEVELOPERS SEEK ON SO AND QUORA

First level	Second level	Definition and Example
Commenting High levels	Add comments	D: general questions about adding comments without mentioning a specific programming language, tool or IDE
		E: *whats the most professional and informative way of commenting code?*
	Versioning comments	D: questioner asks about best practices for comments in code versioning tools like git or svn.
		E: *how to formulate a commit message?*
	Comments example	D: questioner asks for specific examples (funny, helpful, silly) of code comments they have seen.
		E: *What's the least useful comment you've ever seen?*
	Grammar rules	D: ask about following grammar rules in writing comments.
		E: *Should .net comments start with a capital letter and end with a period?*
	Maintain comments	D: general questions about maintaining comments over time.
		E: *Maintenance commenting*
	Other	D: general conceptual questions about code comments
		E: *What's a good comment/code ratio?*
	Syntax & format	D: general questions about syntax and format of comments irrespective of a development environment
		E: *Documentation style: how do you differentiate variable names from the rest of the text within a comment?*
Languages	Adopt other language style	D: questions about adopting the commenting style of another programming language
		E: *Is it a bad practice to use C-style comments in C++ code?*
	Asking for Feature	D: questions regarding whether a feature is supported or not and if not, then how a problem can be solved in the language
		E: *How do I put code examples in .NET XML comments?*
	Asking tool existence	D: Users ask whether there is a tool for a particular programming language to document code
		E: *Is there any specific tool that is used by underscore authors to generate documentation in javascript?*
	Change comment template	D: questions about modification in the template of a specific programming language or environment.
		E: *How to propose some revision to RFC(s) for JSON to account for comments?*
	Process comments	D: questions about processing comments of a particular programming language. Processing includes striping, removing, extracting, cleaning comments
		E: *Remove comment blocks bounded by "#—...—#" in textfile - python*
	Syntax & Format	D: questions about the syntax of comments in a specific programming language
		E: *Commenting in c++?*
	Understand documentation	D: questioner face difficulties with understanding code documentation of function, class, or project
		E: *How to interpret cryptic Java class documentation?*
	Using features	D: user is aware about the feature but does not know how to use the feature
		E: *What are these tags @ivar @param and @type in python docstring?*
Tools	Asking for features	D: question regarding whether a feature is supported or not, or how a problem can be solved with the tool.
		E: *How to properly write cross-references to external documentation with intersphinx?*
	Change comment template	D: questions about modification in the comment template provided by the tool.
		E: *Qt-style documentation using Doxygen?*
	Error	D: questioner needs help with some error or warning received through the tool while writing documentation
		E: *Stylecop doesn't understand <inheritdoc>]*
	Process comments	D: questions about processing comments in a tool. Processing includes striping, removing, extracting, cleaning comments
		E: *Gradle groovy how to keep comments and all formats in XML parser*
	Report Bug	D: questioner reports a (potential) bug.
		E: *Doxygen C# XML comments and generics do not generate links in HTML output?*
	Setup	D: questioner ask about the configuring the tool
		E: *How can I configure GhostDoc to generate comments for attributes on properties?*
	Syntax & Format	D: questioner asks ways to document specific code elements such as class, methods or parts of code in a documentation tool.
		E: *How do I refer to classes and methods in other files my project with Sphinx?*
	Using Feature	D: question regarding how to use a certain feature of the tool.
		E: *How to use @value tag in javadoc?*
IDEs & Editors	Asking for Feature	D: questions regarding whether a feature is supported or how a problem can be solved in the environment
		E: *Android - (Android Studio) - Create similar to TODO, but different COLOR AND NAME?*
	Change comment template	D: questions about modification in the template in the IDE or editor.
		E: *Add different default and custom tags to Visual Studio XML Documentation*
	Process comments	D: questions about processing comments in an IDE or an editor. Processing includes striping, removing, extracting, cleaning comments
		E: *How do I get rid of XXX Auto-generated method stub?*
	Shortcut	D: question regarding how to achieve certain functionality with a shortcut (keyboard)
		E: *Finding Shortcuts in Aptana Studio 3.0 to Comment Code*
	Syntax & Format	D: questions about the syntax of comments in an IDE or editors
		E: *How to comment SQL statements in Notepad++?*
	Tool setup	D: setup a documentation tool in a particular IDE or editor
		E: *How to generate javadoc using ubuntu + eclipse to my project*
	Using features	D: user is aware about the feature but does not know how to use the feature
		E: *How can I get Xcode to show my documentation comments during option-hover?*
Other		D: the questions not belonging to any above category
		E: *Why does Godot receive praise for its fantastic documentation and ease of use from a coding perspective?*

Data Preprocessing, Dataset Availability [2]. To address these reproducibility concerns in our study, we used the tool named *Makar* [3].

Data Extraction. For SO, we used the public API provided by Stack Exchange platform whereas for Quora and mailing lists, we scraped the sources by developing our own crawlers.
Data Preprocessing. The data from Stack Overflow contains HTML, code snippets, links and natural language text. To obtain meaningful results from LDA analysis, it was necessary to clean the data. Figure 1 illustrates the preprocessing steps performed by Makar. The *Transformation* describes various built-in transformations of Makar and *Attributes* shows the list of selected fields (as shown in Table II in the paper) from the sources. The transformations have been set up so that each step produced a new attribute on the data record, allowing us not to lose any information, and retrace every change we made to the data. At the beginning of the case study, it was unclear which combination of question and answer attributes leads to the best result. In this phase, the flexible approach to preprocessing found in Makar helped us to work with our data and try different options efficiently.



Transformation	extract_code	strip_html	string_replace	remove_stopwords	word_stemming
Attributes	- Question | Body	- Question | Body	- Question | Body - Question | Title	- Question | Body - Question | Title	- Question | Body - Question | Title

Fig. 1. All preprocessing steps with the used transformation in *Makar*

Dataset Availability To make our dataset available, we provide a replication package containing raw data, validated data and various other supporting material [4].

REFERENCES

[1] E. Aghajani, C. Nagy, O. L. Vega-Márquez, M. Linares-Vásquez, L. Moreno, G. Bavota, and M. Lanza, "Software documentation issues unveiled," in *Proceedings of the 41st International Conference on Software Engineering, ICSE 2019, Montreal, QC, Canada, May 25-31, 2019*, J. M. Atlee, T. Bultan, and J. Whittle, Eds. IEEE / ACM, 2019, pp. 1199–1210. [Online]. Available: https://doi.org/10.1109/ICSE.2019.00122

[2] J. M. González-Barahona and G. Robles, "On the reproducibility of empirical software engineering studies based on data retrieved from development repositories," *Empirical Software Engineering*, vol. 17, no. 1, pp. 75–89, 2012. [Online]. Available: http://dx.doi.org/10.1007/s10664-011-9181-9

[3] "Makar." [Online]. Available: https://github.com/maethub/makar

[4] "Replication Package." [Online]. Available: https://doi.org/10.5281/zenodo.4470126