Contents lists available at ScienceDirect

The Journal of Systems & Software

journal homepage: www.elsevier.com/locate/jss

An empirical study of COVID-19 related posts on Stack Overflow: Topics and technologies[☆]

Konstantinos Georgiou[a], Nikolaos Mittas[b], Alexandros Chatzigeorgiou[c], Lefteris Angelis[a,*]

[a] *School of Informatics, Aristotle University of Thessaloniki, Thessaloniki, Greece*
[b] *Department of Chemistry, International Hellenic University, Kavala, Greece*
[c] *Department of Applied Informatics, University of Macedonia, Thessaloniki, Greece*

ARTICLE INFO

Article history:
Received 8 January 2021
Received in revised form 8 July 2021
Accepted 2 September 2021
Available online 16 September 2021

Keywords:
COVID-19
Pandemic
StackOverflow
Knowledge-sharing

ABSTRACT

The COVID-19 outbreak, also known as the coronavirus pandemic, has left its mark on every aspect of our lives and at the time of this writing is still an ongoing battle. Beyond the immediate global-wide health response, the pandemic has triggered a significant number of IT initiatives to track, visualize, analyze and potentially mitigate the phenomenon. For individuals or organizations interested in developing COVID-19 related software, knowledge-sharing communities such as Stack Overflow proved to be an effective source of information for tackling commonly encountered problems. As an additional contribution to the investigation of this unprecedented health crisis and to assess how fast and how well the community of developers has responded, we performed a study on COVID-19 related posts in Stack Overflow. In particular, we profiled relevant questions based on key post features and their evolution, identified the most prominent technologies adopted for developing COVID-19 software and their interrelations and focused on the most persevering problems faced by developers. For the analysis of posts we employed descriptive statistics, Association Rule Graphs, Survival Analysis and Latent Dirichlet Allocation. The results reveal that the response of the developers' community to the pandemic was immediate and that the interest of developers on COVID-19 related challenges was sustained after its initial peak. In terms of the problems addressed, the results show a clear focus on COVID-19 data collection, analysis and visualization from/to the web, in line with the general needs for monitoring the pandemic.

1. Introduction

The novel coronavirus disease (COVID-19), declared as a pandemic by the World Health Organization (WHO) on 11 March 2020, has spread worldwide leading to the infection of more than 184 million citizens and the cause of over 3.9 million deaths at the time of this writing. Apart from the detrimental effects on public health, COVID-19 has impacted many aspects of our daily lives causing a tsunami of problems and challenges on economy and society, due to governmental restrictive measures for preventing the overdispersion of the disease (Anon, 2021a).

The global crisis generated by the COVID-19 outbreak has inevitably accelerated the adoption of technological advances and digital products with the aim of mitigating or at least minimizing the short and/or long-term effects of the pandemic. Besides the rapid demand for digitalized platforms elaborating the transformation of homes into places for remote education and work, COVID-19 has led to a steep growth in *Scientific Software Development* (SSD). Generally speaking, SSD refers to the design, implementation and testing of software encompassing knowledge from a specific scientific application domain (e.g. biology, health sciences, mathematics, data science etc.) and used with the primary aim of knowledge acquisition and solving of real-world problems (Kelly, 2015). According to Segal and Morris (2008) SSD is fundamentally different from commercial software since the (usually complex) application domain is not understood by the average developer and for this reason a scientist (domain expert) must be heavily involved in software development. This unique characteristic of relying on domain scientists rather than formally trained software engineers is also highlighted in the report on a series of case studies by Carver et al. (2007). In support of health care professionals who have primarily engaged in the uneven battle against COVID-19, practitioners from interdisciplinary domains have rigorously shifted their interest on the development of software related to a wide variety of applications.

[☆] Editor: W.K. Chan.
[*] Corresponding author.
E-mail address: lef@csd.auth.gr (L. Angelis).

https://doi.org/10.1016/j.jss.2021.111089
0164-1212/© 2021 Elsevier Inc. All rights reserved.

Indeed, an exploration of GitHub activity (Anon, 0000a) shows an explosive interest on COVID-19 with more than 117.000 related software projects. These software endeavors are dealing with multiple aspects of the implications of COVID-19, ranging from websites and applications to trackers and visualization tools. In addition, the European Commission has launched the "*Digital Response to COVID-19*" (Anon, 0000b) initiative providing to practitioners from multidisciplinary scientific domains access to a collection of continuously evolving resources that can be used from public administrations, businesses and citizens to tackle this pandemic. Besides hackathons, events and Connecting Europe Facility (CEF) building blocks opportunities for promoting the knowledge-sharing related to COVID-19 crisis, there is a list of open-source software solutions related to APIs and repositories for integrating data-tracking systems, population surveillance, contact tracing etc., aiming to limit the transmission of COVID-19 and inform citizens about infection epicenters.

Apart from SSD, COVID-19 has brought out the necessity for the collection, aggregation and sharing of massive volumes of accurate, coherent and reliable scientific open data (Anon, 2020, 0000c). To this regard, on 11–12 February 2020, a Global Research and Innovation Forum was organized by the World Health Organization (WHO) with the participation of experts and funders from 48 countries aiming to the assessment of the level of knowledge, identification of gaps and acceleration of collaborative and funded research in alliance to the battle against COVID-19 (Anon, 0000c). In addition, on 18 March 2020, WHO and partners launched the "*Solidarity trial*" focusing on the accumulation of data from all over the world in order to identify effective treatments for COVID-19. In parallel, the EU Commission has launched a manifesto with the aim of promoting the open access of generated results, scientific knowledge, data sharing and distribution of products and services by public and private stakeholders, institutions and even individuals funded from EU research grants related to the COVID-19 pandemic (Anon, 2021b).

All the aforementioned synthesize a valuable source of knowledge-sharing that can be used from scientists across interdisciplinary domains such as medical science, epidemiology, biology, pharmacology, bioinformatics, statistics, data science etc. for promoting research regarding the causes, dynamics and consequences of this phenomenon from different perspectives.

Despite the importance of SSD in research progress, it has been reported that domain scientists are less trained to develop software efficiently (Wilson, 2006; Arvanitou et al., 2020; Nguyen-Hoan et al., 2010). Thus, we posit that domain scientists are expected to often consult knowledge-sharing forums to mine solutions on SSD problems. *Stack Overflow* (SO), which has more than 20 million posted questions, approximately 40 million answers and 13 million users, is probably one of the most popular question and answer (Q&A) forums, in which developers, who face problems, seek help and advice from other members of the community. Note that the term "developers" does not necessarily refer only to software professionals.

Due to the abovementioned considerations, the SO ecosystem has attracted the interest of the research community in order to investigate its characteristics and evolution under multiple perspectives. Despite the fact that previous work on SO provides a significant body of knowledge on a wide variety of aspects related to Q&A forums, in this study, we focus on the examination of SO from a different perspective. In particular, the aim of the current paper is to investigate whether, an unprecedented global health crisis rather than technological challenges themselves, has triggered the initiation of knowledge-sharing about problems in the development of COVID-19-related software. From now on and throughout the paper, we use the term "COVID-19 software" to indicate any software that addresses direct or indirect problems related to COVID-19, including data collection and analysis development of applications and web platforms to report and visualize COVID-19 related information, the use of forecasting techniques to predict aspects of the pandemic, etc.

This study constitutes an expanded research of the work undertaken by Georgiou et al. (2020), which combines several of the aforementioned methodologies to explore the impact of the COVID-19 pandemic to software development ventures. In our previous paper (Georgiou et al., 2020), we conducted a preliminary study examining the knowledge-sharing activity in SO covering only the first period of the outbreak (January 26th–April 1st). The current study provides a thorough investigation of the phenomenon through the collection and analysis of a significantly enriched dataset covering a wider timeframe (January 26th–October 28th). Additionally, the methodological framework is also expanded by setting specific research goals in order to gain better insights regarding the responsiveness and general effects of this unprecedent health crisis in the SO knowledge-sharing activity. The motivation of our work was based on the empirical evidence that the COVID-19 pandemic has raised the need for even faster adoption of recent technological advances and the rapid growth of dedicated COVID-19-related software solutions. In particular, the goals of the paper are:

***(g1)* research field analysis**: As a first step for an overview of knowledge-sharing related to COVID-19, we aim at identifying the starting point of users' activity and investigate the evolution trend over time trying to infer about potential reasons for fluctuations throughout the study period. To develop a more comprehensive understanding, we analyze specific features of the posted questions to derive meaningful conclusions about the dynamics of the examined phenomenon, since they are considered key factors for measuring activity and quality in SO (Anderson et al., 2012b). The analysis could be beneficial, since it can act as a pointer to whether a sufficient body of knowledge has been formed in the SO community related to specific challenges in COVID-19 software development enabling its safe re-use from other stakeholders.

***(g2)* identification of technology advances and associated problems**: At a second level, we adopt a more technological perspective. In particular, we first aim at identifying the most prominent technologies adopted for fighting COVID-19. Moreover, we investigate whether there are technological problems that are more difficult to be resolved by SO community. Such information can be useful to a wide range of stakeholders that are interested in developing COVID-19 related software and products, in the sense that it can unveil possible needs for training in specific skillsets. Secondly, we aim to dig further into problems that practitioners face while developing scientific software. The analysis of the textual content of questions in SO can not only reveal valuable semantic insights related to the purpose of a post (Anderson et al., 2012b) but also help researchers in reusing well-established strategies rather than reinventing the wheel.

The rest of the paper is organized as follows: In Section 2, we present related work which is necessary to explore the background of our research and basic definitions. In Section 3, we present some basic background notions and define the goals of study and research questions adherent to these goals, while in Section 4, we analyze the methodology framework we applied and explain the separate steps conducted. In Section 5, we present the findings and discuss the produced results and explanations while Sections 6 and 7 serve as a discussion of the findings and the usefulness of the current study to the wider scientific community and a discussion of relevant threats to validity, respectively. Finally, Section 8 offers some closing remarks and conclusions about the undertaken research.

2. Related work

In this section, we present recent literature relevant to this study. Our purpose is to signify the importance of Q&A communities, and SO in particular, in information extraction as well as the multifaceted scopes of the undertaken research. In general, the existing literature explores various subjects such as the identification of user characteristics and activity (Rosen and Shihab, 2016), gamification/reputation mechanisms (badges etc.) (Papoutsoglou et al., 2020), tagging activity regarding the technical aspects of research, factors that influence the timeframe of a question receiving an answer (Mamykina et al., 2011), the semantic information hidden in posts (Chen and Xing, 2016; Ye et al., 2017), the reasons behind questions (Asaduzzaman et al., 2013) and the exploration of discussion topics (Beyer et al., 2020).

The primary purpose of Q&A communities is the collaboration and opinion exchange among individuals of different expertise and knowledge regarding various topics. Their flow of information relies on the wisdom of the crowd, the rapid social interactions and users demonstrating their technical and conversational capabilities. Self-presentation can be crucial for engagement in such communities (Raban, 2009). Moreover, Q&A communities and SO in particular, are quite timely in detecting and highlighting emerging technological trends (Chen and Xing, 2016; Ye et al., 2017). In SO, answering patterns indicate that posted questions receive a response in a relatively short time span (Asaduzzaman et al., 2013; Wang et al., 2018b, 2020). A question may remain unanswered for specific reasons that typically involve a vague description or the absence of code examples to support the textual content (Wang et al., 2018b).

User reputation in SO is highly important, affecting the probability of a post receiving answers (Bosu et al., 2013). Bazelli et al. (2013) categorize users based on their personality and associate extroverted users with increased reputation and answering activity. Similar experimentations have been conducted in another study (Mamykina et al., 2011), where the community dynamics and the median time of answering a question predict the added value of a question to the website.

Emphasis is also given to the semantic information of the questions and the purpose of their creation. Frequently, a post concerns a technical problem that will require a solution or guidelines for the implementation of software (Linares-Vásquez et al., 2013). In other instances, questions concern inquiries about changes between software versions, errors and gaps in code maintenance or unexpected setbacks in development (Beyer and Pinzger, 2014). Useful insights that reveal the purpose of a question is the usage of inquiry words (e.g. "*why*"," *how*") or the inclusion of verbs related to a purpose (e.g. "*try*") (Allamanis and Sutton, 2013; Treude et al., 2011).

Published questions and answers can differ significantly in terms of quality. Outdated or misguided questions can be approved and answered by users, creating confusion and erroneous or suboptimal software implementations. Filtering and detecting published content, while isolating low-quality and promoting high-quality posts is vital for the sustainability of such communities and has been explored by several studies (e.g. (Ortega et al., 2014; Neshati, 2017)). To ensure that competent answering is rewarded, SO is employing gamification mechanisms, distributing "badges" to esteemed responders and encouraging revisions and edits to posted questions and answers as well as the rapid answering of posts (Papoutsoglou et al., 2020; Wang et al., 2018a).

Apart from semantic differences and answering patterns, there are several studies that attempt to classify SO posts in thematically relevant topics. These topics concern different aspects of technological knowledge and can either focus on a specific technological domain or a wider range of fields. Wang et al. (2013) categorize posts in specific topics related to code generation (User Interfaces, Web Documents etc.) To that end, they utilize the Latent Dirichlet Allocation (LDA) (Blei et al., 2003) algorithm. In general, the LDA algorithm constitutes a robust method for topic identification and is leveraged in several topic related studies. Barua et al. (2014) explore the evolution of discussion topics in time and address the possibility that answers in a specific discussion thread can spark interest for posts belonging to different topics.

Several studies investigate the involved technological areas, to better grasp relevant obstacles and inquiries. For instance, given the rise of smartphones mobile development in SO is continuously analyzed by researchers to extract topics (Rosen and Shihab, 2016; Linares-Vásquez et al., 2013; Villanes et al., 2017). Beyer and Pinzger (2014) also categorize Android development questions but delve deeper in the inquisitive nature of posts and the problem-solving processes that accompany their answers. Along the same lines, dominant topics are investigated in software maintenance and legacy code (Ahmed and Bagherzadeh, 2018), the usage of web frameworks and APIs (Venkatesh et al., 2016) as well as security and privacy (Yang et al., 2016). Zou et al. (2017) emphasize on the non-functional attributes of software (e.g. scalability, maintainability) to pinpoint potential shortcomings in software lifecycle management. Johri and Bansal (2018) introduce the concept of topic impact and popularity, utilizing dedicated metrics that compute the inter-post relationships of topics over time. In another study, *Topic Shifting* (Gruetze et al., 2016) is defined as the variations in the usage of specific tags for discussion topics as software and ICT technologies evolve. Similar practices are employed by Shao and Yan (2017) with the intent of utilizing the topic distribution of a question for recommending the most appropriate users that can provide well-documented and thorough answers. Finally, Chen et al. (2019) organize synonym tag communities in concepts and perform hierarchical clustering to discover relationships between cross disciplinary tags that are used in multiple subjects of discussion.

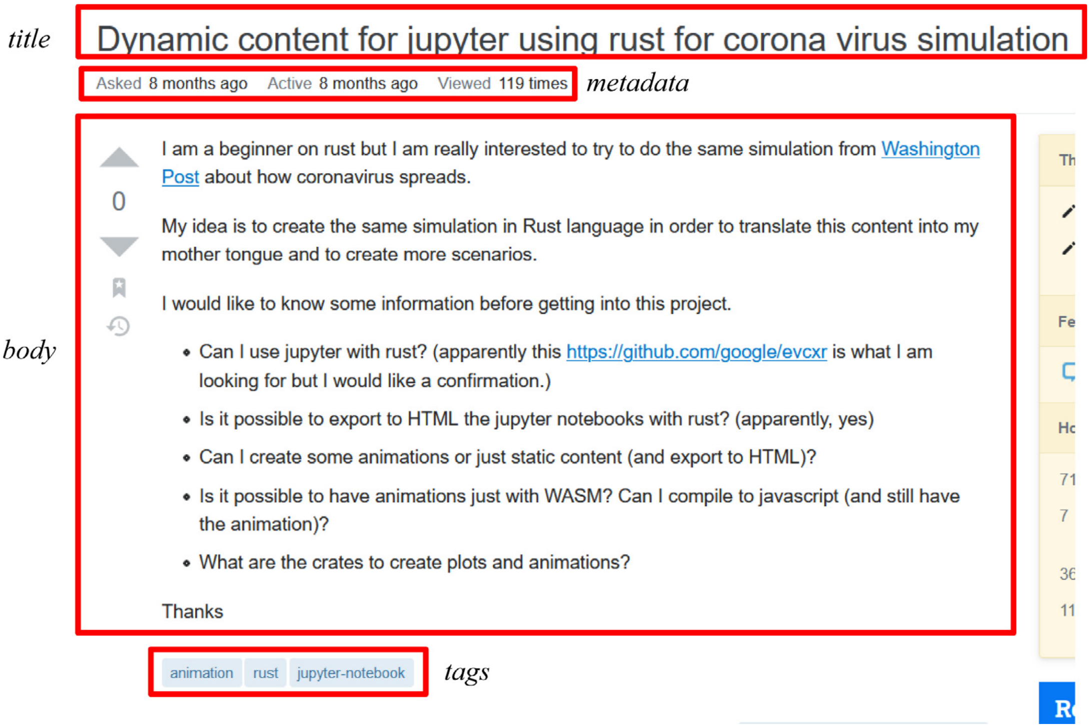
Beyond the use of topic extraction methodologies, networks of co-occurring tags have been employed to profile tags and uncover user activity around them (Chen and Xing, 2016). Co-occurring tag networks are also employed (Chen and Xing, 2016; Westwood et al., 0000; Georgiou et al., 2019) to represent tags as separate clusters that express different areas of expertise and knowledge as well as reputation.

3. Background information and research questions

3.1. Background information

The basic entity of information in our study is SO posts related to COVID-19, which contain questions and answers. An illustrative example with some key elements highlighted is presented in Fig. 1. The main part of a post is the question being posted, with the *title* being a brief description of the question's content and the *body* including more detailed information. Apart from this, each question contains other useful *metadata* such as *views*, *votes* and the question's *creation date*. A question can also have a certain number of answers that provide solutions or guidelines.

In addition, each question post is labeled with *tags* providing straightforward information about the technologies related to the topic of discussion. Although tags are considered as a starting point for investigating the technological issues and difficulties that developers are facing (Allamanis and Sutton, 2013; Blei et al., 2003), the tagging mechanism is a user-defined process that has led, in turn, to the "*tag explosion*" problem (Blei et al., 2003). To overcome this limitation, which constitutes a significant inhibitor for tracing prominent technologies related to COVID-19 software



Dynamic content for jupyter using rust for corona virus simulation

Asked 8 months ago Active 8 months ago Viewed 119 times

0

I am a beginner on rust but I am really interested to try to do the same simulation from Washington Post about how coronavirus spreads.

My idea is to create the same simulation in Rust language in order to translate this content into my mother tongue and to create more scenarios.

I would like to know some information before getting into this project.

- Can I use jupyter with rust? (apparently this https://github.com/google/evcxr is what I am looking for but I would like a confirmation.)
- Is it possible to export to HTML the jupyter notebooks with rust? (apparently, yes)
- Can I create some animations or just static content (and export to HTML)?
- Is it possible to have animations just with WASM? Can I compile to javascript (and still have the animation)?
- What are the crates to create plots and animations?

Thanks

animation rust jupyter-notebook

Fig. 1. Example of COVID-19 related post.

development, we make use of a *technology reference hierarchy* (Fig. 2) that categorizes each post on broad *Technology Classes* (TCs) based on specific *technologies* found in the set of tags.

More precisely, the basis for the exploration of key technologies is the construction of a lexicon constituting an assembly of technologies retrieved from the yearly Developer Surveys conducted by SO during the period 2014–2020. Our preference on the SO Developer Survey (Stack overflow developer SURVEY 2020, 0000) over other similar lexicons found on the web is due to the fact that SO is one of the most prominent and prestigious knowledge-sharing communities synthesizing an immediate and precise source of technological topics that are discussed by professionals. The SO surveys provide extensive coverage of topics and essentially revises the technical and ICT related content of the community, with 65.000 developers and experts providing feedback about their activities and experiences. Moreover, they leverage a thorough categorization of technologies by classifying them to broad classes based on the technological aspect they address.

To this regard, the lexicon can be perceived as a technology reference hierarchy consisted of two separate tiers (Fig. 2). The First Tier is comprised of seven distinct TCs describing more generic technological aspects, whereas the Second Tier contains 182 specific technologies. Concerning the first level of hierarchy, the **Languages**[1] category is associated to programming languages, such as *python, javascript, r* etc. **Web Frameworks** correspond to special purposes, self-packaged environments oriented to the building and deployment of front-end and back-end infrastructures (e.g. *angular*). The **Big Data/ML** category contains technologies that are operated for streaming and processing large volumes of data and train machine learning models for specialized purposes. **Developer Tools** and **Collaboration Tools** refer to well-known *Integrated Development Environments* (IDEs) prominently employed for writing code and software sharing and automation testing suites, respectively. Operating systems, virtual environments and hosting services are classified to the **Platforms** category, whereas database management systems and database handling suites are contained in the **Databases** category. At this point, we have to clarify that question posts containing technologies belonging to multiple TCs will be classified in more than one TCs.

While the set of *tags* provide information about the technological aspect of a question, the textual information (*title*, *body*) is an ample source of knowledge regarding the *topic* of discussion, as presented by other similar studies (Beyer and Pinzger, 2014; Blei et al., 2003). We conceive the *topic* of discussion as a subset of SO posts classified under a common thematic axis, characterized by specific words and terms. Thus, a *topic* represents a thematic area of posts expressed through language and semantics (e.g. posts asking about "Creating COVID-19 Simulations") and not by exploring technology related elements such as *tags*. In addition, a question post may contain a mixture of topics since the textual information of the *title* and *body* fields can correspond to multiple thematic axes (Barua et al., 2014).

3.2. Research questions

The main pillar of this study, as mentioned in Section 1, is to investigate the phenomenon of software development in the light of COVID-19 era and its implications to knowledge pathways as expressed in a well-known Q&A forum such as SO. However, we do not focus on the produced results in terms of COVID-19 related software products and services. Instead, we chose to examine the trends in technological advances related to SSD and general purposes and the risen technological barriers and practical obstacles encountered by developers during the implementation of COVID-19 software. To achieve our objectives, we formulate the following research questions (RQs) aligned to the two general goals presented in Section 1 and within the aforementioned background definitions.

[1] For the rest of the paper, the TCs of the First Tier are highlighted by bold fonts

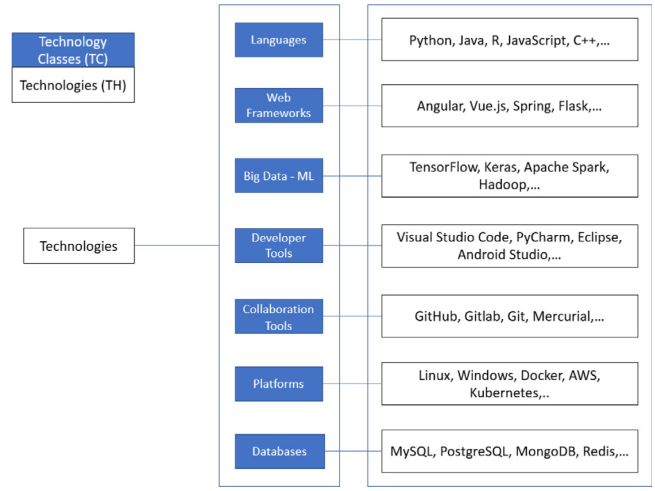

Technology Classes (TC)
Technologies (TH)
Technologies
Languages
Web Frameworks
Big Data - ML
Developer Tools
Collaboration Tools
Platforms
Databases
Python, Java, R, JavaScript, C++,...
Angular, Vue.js, Spring, Flask,...
TensorFlow, Keras, Apache Spark, Hadoop,...
Visual Studio Code, PyCharm, Eclipse, Android Studio,...
GitHub, Gitlab, Git, Mercurial,...
Linux, Windows, Docker, AWS, Kubernetes,..
MySQL, PostgreSQL, MongoDB, Redis,...

Fig. 2. Technology reference hierarchy for categorization of SO question posts.

[$\mathbf{RQ_{1.1}}$] *Did the evolution of COVID-19 pandemic trigger corresponding knowledge-sharing activity in SO and how has this phenomenon evolved over time?*

[$\mathbf{RQ_{1.2}}$] *Which are the characteristics of COVID-19 knowledge-sharing activity in SO?*

The first RQ ($\mathrm{RQ}_{1.1}$) aims to investigate, whether the critical circumstances caused by the outbreak of a worldwide healthcare crisis have motivated developers to actively get involved in software development that, in turn, would result in seeking advice and help about technological barriers during the process in well-known knowledge-sharing communities. The analysis of users' post activity associated to COVID-19 and tracking of its evolution over time will provide insights related to the body of knowledge created during the examined period and the identification of potential peaks and falls in activity. Finally, as the scientific community is still in an early phase in countering and eradicating the pandemic and there is increasing interest in adapting cutting-edge digital technologies to study and address problems caused by COVID-19, we believe that the performance of knowledge-sharing communities in providing open access support of high-quality standards without delay deserves investigation ($\mathrm{RQ}_{1.2}$).

Apart from the research field analysis on the examined topic (***g1***), the second goal of this study (***g2***) is to gain insights related to knowledge-sharing in COVID-19 related posts. To this regard, we formulate the following RQs:

[$\mathbf{RQ_{2.1(a)}}$] *Which technologies are more popular in COVID-19 software development and how are these associated to each other?*

[$\mathbf{RQ_{2.1(b)}}$] *Which technologies present more difficulties in COVID-19 software development?*

[$\mathbf{RQ_{2.2}}$] *Which topics are more popular in COVID-19 software development?*

In a relatively short time span, the pandemic has spread alarmingly fast resulting in continuously growing technological challenges in COVID-19 software development (George et al., 2020; Brem et al., 2021). This is proven by the growing efforts of prestigious organizations and alliances, including WHO and the EU Commission, in developing cutting-edge solutions for battling the pandemic. In this joined effort, the role of technology specialists who exploit current technological means to support the epidemiological, biological and data related aspects of these initiatives is vital (Kumar et al., 2020; Vaishya et al., 2020). Thus, the second goal of the current study is two-fold. To this regard, we base our approach on both the technological insights expressed by SO tags and the semantic structures of questions adopting the distinction between the technological content and the set of reasons questions are asked (Beyer et al., 2020). More specifically, Beyer et al. (2020) point out that *problem categories* refer to "*the topics or technologies that are discussed*" and they are expressed by the SO tagging system, providing users with a straightforward mechanism for labeling their posts to specific technological aspects. In contrast, *question categories* represent "*the kind of information requested in a way that is orthogonal to any particular technology*" (Beyer et al., 2020).

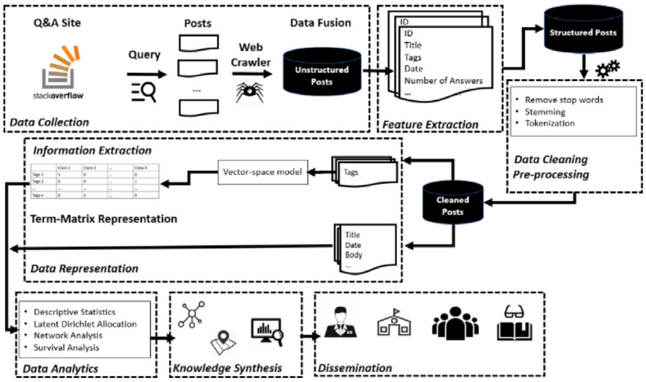


Fig. 3. Research approach of the study.

Based on these definitions, in $\mathrm{RQ}_{2.1(a)}$, our aim is to identify broad technology classes and related prominent technologies that have been adopted in COVID-19 software development and explore whether there are interconnected technologies raising a subject of debate in SO community. The existence of such popular and interconnected technological advances would provide certain directions regarding the demand for cutting-edge technological skillsets. In addition, the identification of a potentially higher level of difficulty for specific classes of technologies ($\mathrm{RQ}_{2.1(b)}$), would bring to the surface specialized needs for fostering the training of developers in order to fulfill their competence gaps related to COVID-19. Subsequently in $\mathrm{RQ}_{2.2}$, our aim is to discover the main topics of discussion in COVID-19 posts and investigate the purposes of different posts. The identification of such topics would certainly provide detailed insights to the interests and activities of developers during the COVID-19 era, uncovering the core fields that support SSD and projects related to the pandemic. This, in turn, would give a clear picture of future focus from individuals that wish to enhance their skillsets in order to further explore their capabilities in similar projects and contribute to the COVID-19 software development ecosystem.

4. Methodology

In this section, we present the approach followed in this study to meet the general goals by providing answers to the posed RQs (Section 3). An overview of the methodology is presented in Fig. 3 that can be described as an approach consisting of seven phases namely: (*i*) *data collection*, (*ii*) *feature extraction*, (*iii*) *data cleaning and pre-processing*, (*iv*) *data representation*, (*v*) *data analytics*, (*vi*) *knowledge synthesis* and (*vii*) *dissemination* of the extracted results.

4.1. Data collection

Based on the motivating idea of the current study, we decided to utilize SO as the main data repository to identify and extract posts discussing technological issues during the development of software related to the COVID-19 pandemic. To this regard, we followed a semi-automated search strategy by formulating a quite broad search string encompassing synonyms of the coronavirus term (first round of data collection process). The final search string, defined through an iterative approach after trial searches, used the following terms: *“coronavirus”* OR *“covid*”* OR *“corona-virus”* OR *“sars-cov”* OR *“2019-ncov”*. The data collection process was completed on 28th of October 2020 resulting into the identification and extraction of 2719 questions, from which we excluded questions that had been marked as *“Closed”* by the platform, indicating that either they were containing low quality content or had been asked in a different manner in other posts[2] (Correa and Sureka, 2013; Ponzanelli et al., 2014; Ahasanuzzaman et al., 2016).

At the second round of the data collection process, the first two authors independently read the body of each post with the aim of identifying and filtering out posts encompassing a term of the predefined search string without, however, seeking for any advice related to COVID-19 software development. To this regard, no conflicts were identified in the characterization and removal of posts. Below, we indicatively present an example of post that was filtered out during the second round of the data collection process.

> I hope that you are all safe from the outbreak of coronavirus and I pray for the heroes who are in the frontlines fighting against this epidemic. I’ve been facing a problem with WooCommerce order’s emails layout … . When the website language is “English” the email content is perfectly delivered! However, when the website language is “Arabic” …. the content position is viewed like this: …. Using Gmail and Outlook Apps on both Android and IOS, using email customizer to edit the width & the height of the emails, none of them could solve the problem. …..

After the filtering process, the final dataset contained 2213 question posts. The collection of question posts and their features was conducted by utilizing a web scraper built in Python based on the Selenium package (Selenium, 0000).

4.2. Feature extraction

The data collection process returned a set of semi-structured web documents comprised of COVID-19 related posts covering the examined period. The foundation for building our retrieval methodology was the definition of a question post as a self-contained entity in the SO ecosystem, containing a rich source of information from which meaningful features can be extracted. More specifically, a *Question Post* (QP) is defined as a multi-element tuple

$$QP = (id, ti, b, tg, d, vi, vo, na, c, sn, ad) \tag{1}$$

where each element is briefly described in Table 1. The implemented web crawler scrapped each post independently and retrieved necessary metadata storing them in separate lists. The produced lists were then, unified into a single database that was used in the later stages of the proposed approach.

Table 1
Extracted features from SO question posts.

Feature	Description
id	Identification number of the question
ti	Title of the question
b	Body of the question
tg	List of tags associated to the question
d	Creation date of the question
vi	Question views
vo	Question votes
na	Number of answers
c	Number of comments
sn	Indicator variable of code snippet (absence/presence)
ad	Creation date of the first answer

4.3. Data cleaning & pre-processing

The final dataset of question posts was subjected to necessary pre-processing and cleaning procedures, to ensure data quality and remove unwanted noise. The textual features (i.e. *title* and *body*) were firstly transformed to lowercase, while punctuation marks were removed along with URLs, special characters and delimiters. Additionally, each post was tokenized and stemmed, whereas stop words and whitespaces were removed. For these purposes, the NLTK (Anon, 0000e) python package was utilized.

4.4. Information extraction and data representation

Given that textual features (*title, body, tags*) provide useful information, dictating the related technologies and purposes of a post (Section 3), the next step involved the transformation of semi-structured data into an appropriate representation format in order to derive meaningful conclusions. To this regard, we made use of *Text Mining* (TM) techniques to leverage the textual information to their full extent.

4.4.1. Tags feature

As previously mentioned in the Introduction, *g2* aims at identifying broad technology classes and specific technologies that serve as catalysts for COVID-19 software development. Given that the tagging mechanism provides certain directions about the technological aspects of a question post, we give particular emphasis on the *tags* field. On the other hand, despite the fact that this labeling mechanism presents some merits regarding the technological content of a post, it also poses certain practical challenges due to the detailed and broad list of user-created tags (Beyer et al., 2020). In order to provide straightforward answers to $RQ_{2.1(a)}$, we relied on the lexicon defined in Section 3 in order to discard tags that constituted noise and keep only those relevant to the scope of our study.

Having the lexicon as a basis of analysis, we subjected the list of the derived SO tags found in question posts to several transformations in order to ensure compatibility and remove redundant synonym terms. For example, tags referring to different versions of the Python programming language (e.g. *“python 3.6”, ”python 2.7”*) were simply reverted to *“python”*. Similar pre-processing steps were followed for other tags referring to identical versions of software or different implementations of a specific framework.

The next step involves the matching of *tags* found in question posts on the basis of the predefined technology hierarchy, so as to represent each question post an appropriate format. This matching process facilitates the representation of question posts through a multi-dimensional *Vector Space Model* (VSM) comprised of Boolean terms. The matching is conducted both for the TCs of the First Tier as well as the terms of the Second Tier.

We showcase a representative example of a question post categorization via the proposed matching process based on the

[2] Closed questions cannot be answered but can be edited to make them eligible for reopening.

predefined technology hierarchy by examining the question post of Fig. 1. As *"rust"* is a tag belonging to the broad **Languages** TC, in the first stage of the matching process, the question is categorized into this particular class. The derived vector representing the question post in VSM has the form of $\{1, 0, 0, 0, 0, 0, 0\}$, where zeros and unities indicate the absence or presence of a specific tag related to the seven broad TCs, respectively. In this case, the single "1" indicates that the post under examination is matched only to the **Languages** category of the First Tier. In the second stage, we follow a similar approach for representing question posts based on information related to specific technologies (182 in total) of the Second Tier of the predefined lexicon.

4.4.2. Title and body features

Regarding the investigation of the semantic structure hidden in question posts, we exploited the textual information extracted from the *title* and *body* fields for the collection of posts, since it provides an overview of the purposes behind its posting. More specifically, given that the *title* serves as a self-contained and brief presentation of a question, we can conclude that merging the *title* with the *body* can be a potent indicator of the subject of a post and the meaningful semantics utilized for its expression (Ponzanelli et al., 2014; Ahasanuzzaman et al., 2016). This merging was done in order to meet the objective of $RQ_{2.2}$ aiming at the identification of popular topics in COVID-19 software development posts.

4.5. Data analysis

The next phase of the methodology involves the application of appropriate statistical analysis methods for accomplishing the goals of the current study through the examination of the posed RQs (Section 3). Table 2 provides an overview of the goals, the associated RQs along with the extracted features from posts and the data analysis methods employed for each RQ. Regarding the first goal (***g1***) and the corresponding RQs ($RQ_{1.1}$ and $RQ_{1.2}$), we made use of SO metadata features and appropriate univariate descriptive statistics and visualization techniques for investigating the distributions of both qualitative and quantitative characteristics of COVID-19 related posts. Especially, for $RQ_{1.2}$, we made use of appropriate statistical hypothesis testing procedures to examine whether the observed phenomena can be generalized to the population. More specifically, the *chi-square test of independence* was performed in order to assess, whether there was noted a statistically significant association between two categorical variables, whereas the measure of *phi* (φ) was used to calculate the effect size. For count variables, the non-parametric *Mann–Whitney* test was used to examine potential differences in the distributions of two independent populations, whereas the r statistic utilizing the z value of the test and the total number of observations was undertaken for calculating the effect size.

Concerning the second goal of the study (***g2***) and the corresponding RQs ($RQ_{2.1(a)}$, $RQ_{2.1(b)}$ and $RQ_{2.2}$), we performed appropriate multivariate statistical methods based on the specific needs of the posed RQs and the type of the available information extracted from the collection of question posts. More specifically, for $RQ_{2.1(a)}$, the primary objectives were (*i*) to identify both broad TCs and prevalent technologies leveraged for COVID-19 software development and (*ii*) explore potential interconnections among them. In order to meet these objectives, we explored the distributions of the extracted tags for each TC of the hierarchy presented in Fig. 2. The rationale behind the choice of examining the distributions for each TC separately, instead of simply analyzing the set of *tags* extracted from all question posts, was the fact that this strategy would be beneficial in the identification of prominent technologies that refer to specialized skillsets fulfilling different purposes regarding COVID-19.

The investigation of interconnections between technologies was based on the VSM representations for the First and Second Tiers of Technology hierarchy (Fig. 2). The rationale behind this approach was based on the fact that the predetermined lexicon is divided into two levels representing broad TCs (First Tier) that are further divided into specific technologies (Second Tier). Thus, conducting an analysis on both levels of the hierarchy will facilitate the general comprehension of the interactions between different broad TCs and provide more detailed insights on different associations between specific technologies. To this end, we evaluated the co-occurrences of tags in questions, whereas the adoption of *Graph Theory* methods contributed to the identification of clusters with interconnected technologies. More specifically, the co-occurrences of TCs in the set of question posts were used as input for the construction of networks, where each node represents a TC from the First Tier of the hierarchy and edges connecting two nodes represent the total number of co-occurrences between pairs of TCs.

Graph Theory: To investigate potential patterns among specific technologies of the Second Tier, there was a need to make use of an appropriate metric that would be able to capture the strength of the associations between them. For this reason, we followed an approach similar to the one proposed by Cui et al. (2010), exploiting *Association Rules Graphs* (ARG) for investigating the trend of evolution in collaborative tagging systems. The notion of a tagging system is consistent to the framework of our study, since each post carries a set of k tags (from one up to five) in order to label its technological content. Based on this idea, Cui et al. (2010) proposed the visualization of tags and their *associations* through the construction of an ARG.

An ARG is evaluated based on information derived from three metrics, known as (*i*) *frequency*, (*ii*) *support* and (*iii*) *confidence*. Given two *tags*, namely tag_i and tag_j, the frequency ($freq(tag_i)$) of tag_i is computed by summing up the total number of occurrences in the set of question posts. The *support* metric ($supp(tag_i, tag_j)$) quantifies the number of co-occurrences of tag_i and tag_j, whereas *confidence* ($conf(tag_i \rightarrow tag_j)$) expresses the conditional probability of tag_i occurring in a post that has been already tagged by tag_j, given that $freq\,(tag_i) < freq\left(tag_j\right)$. Eq. (3) provides the formula for the evaluation of confidence for tag_i and tag_j, given that $freq\,(tag_i) < freq\left(tag_j\right)$

$$conf(tag_i \rightarrow tag_j) = \frac{supp(tag_i, tag_j)}{freq(tag_i)} \tag{2}$$

Based on the abovementioned definitions, an ARG can be graphically displayed via a directed graph $G = (V, E)$, where V and E represent the set of vertices and edges, respectively. In this study, each tag (tag_i) is visualized by a specific node with an associated weight (w_{tag_i}) representing its frequency ($freq(tag_i)$). In addition, a directed edge is constructed for each pair of tags $\{tag_i, tag_j\}$ that co-occurred in the set of question posts satisfying the condition $freq(tag_i) < freq(tag_j)$, whereas the edge is also weighted by the confidence metric ($conf(tag_i \rightarrow tag_j$). Finally, we have to clarify that an ARG was constructed for each TC taking into consideration the set of *tags* belonging to a specific TC along with their connections with *tags* from other TCs, so as to investigate both internal and external patterns of prominent technologies.

Survival Analysis: After the identification of prominent technologies and their interconnections ($RQ_{2.1(a)}$), the interest is now focused on the exploration of the level of difficulty raised by specific TCs in COVID-19 software development ($RQ_{2.1(b)}$). To this regard, we based our inferential process on information extracted from the distributions of the time elapsed for a post to receive its first answer (Rosen and Shihab, 2016; Ortega et al., 2014), rather than the time elapsed between the posting of a question

Table 2
Research goals, research objectives and extracted features from SO posts.

Goal	Research question	Features	Data analysis method(s)
Research field analysis (*g1*)	[$\textbf{RQ}_{\textbf{1.1}}$] *Did the evolution of COVID-19 pandemic trigger corresponding knowledge-sharing activity in SO and how has this phenomenon evolved over time?* [$\textbf{RQ}_{\textbf{1.2}}$] *Which are the characteristics of COVID-19 knowledge-sharing activity in SO?*	SO metadata (d, vi, vo, na, c, sn)	Descriptive statistics, Statistical Hypothesis Test (chi-square test of independence, Mann–Whitney)
Identification of technology advances and associated problems (*g2*)	[$\textbf{RQ}_{\textbf{2.1(a)}}$] *Which technologies are more popular in COVID-19 software development and how are these associated to each other?*	List of tags (tg)	Graph Theory (Association Rule Graph)
	[$\textbf{RQ}_{\textbf{2.1(b)}}$] *Which technologies present more difficulties in COVID-19 software development?*	Time elapsed until the first answer to be posted ($t = d - ad$)	Survival Analysis (Kaplan–Meier curves)
	[$\textbf{RQ}_{\textbf{2.2}}$] *Which topics are more popular in COVID-19 software development?*	Textual information (t, b)	Latent Dirichlet Allocation

and its accepted answer. The reason for this choice was the fact that the percentage of posts that received an accepted answer is usually significantly lower compared to the percentage of posts that received at least one answer (Rosen and Shihab, 2016), since only the original user who posted a question can mark it as accepted, which is not a required action (Ortega et al., 2014) or the user may forget to accept an answer (Ortega et al., 2014). Although the main idea is to examine the distribution of time it takes for a question to receive the first answer, we decided to follow an alternative approach introduced by Ortega et al. (2014) concerning the analysis of the duration variable.

More specifically, *Survival Analysis* (Kleinbaum and Klein, 2012), a well-known time-to-event statistical methodology examining the distribution of the duration from a starting time origin to an endpoint of interest, was adopted. Our preference to this specific approach rather than other traditional statistical methods is based on the fact that Survival Analysis takes into account not only observations experiencing the event of interest but also cases for which the predefined terminal event has not been occurred over the examined follow-up period. Survival Analysis has often been used in medical research where the terminal event (such as cure or death) has not occurred for a number of patients up to the time point of the study.

Describing briefly, in our case, the variable of interest is defined as the *time elapsed until the first answer to be posted* (terminal event). Despite the reputation of SO in providing timely and effective solutions satisfying high-quality standards, there is also a subset of question posts that have not received an answer until the end of the study period that is the completion date of the data collection process. These unanswered posts are defined as *censored* observations in SA terminology, representing cases for which there is available information that should be taken into consideration, when analyzing the performance of users' activity in terms of their responsiveness. This is, in fact, the main advantage of SA over other traditional time-to-event analysis statistical methods, since the latter methods completely ignore such type of information related to censored cases. Summarizing, the time elapsed for a given answered post was calculated by subtracting the timestamp of the first received answer from the creation timestamp of the post. As far as the set of censored cases concerns, the time elapsed for these unanswered question posts was evaluated by subtracting the final date of the data collection process (October 28, 2020) from the creation timestamp of the post.

Based on the above considerations, for the formal representation of the general principles of Survival Analysis, the time elapsed until the first answer to be posted can be considered as a positive random variable, denoted by T. The survival function $S(t)$ that evaluates the probability that the time elapsed until the first answer to be posted is longer than t is defined as

$$S\left(t\right) = P\left(T > t\right) \tag{3}$$

For the evaluation of the survival function $S(t)$, we made use of a well-known non-parametric statistical technique, known as the *Kaplan–Meier* (K–M) method (Kaplan and Meier, 1958), which involves the estimation of probabilities of occurrence of event (post of the first answer) at a certain point of time t_i and the multiplication of these successive probabilities by any earlier computed probabilities. The K–M estimation of $S(t)$ at a certain time point t_i is given by the following recurrent equation

$$\hat{S}\left(t_i\right) = \prod_{j=1}^{i}\left(1 - \frac{d_j}{n_j}\right) \tag{4}$$

where, d_j the number of question posts received a first answer and n_j the number of posts waiting for the first answer at an earlier time point t_j. Beside this recurrent formula, the K–M method is augmented with a powerful visualization tool, namely the K–M *curve*, that provides straightforward interpretation of the duration of the time needed until the first answer to be posted on the basis of the distribution shape. More precisely, a steep curve indicates short elapsed times until the first posted response, which practically means problems that are less difficult to get resolved. In contrast, flat curves demonstrate longer times before the first answer to be posted and thus, issues that deserve more effort and maybe higher level of expertise.

Latent Dirichlet Allocation: Concerning RQ$_{2.2}$, the aim was to detect semantic patterns in question posts leveraging the corpus of titles and bodies of question posts. This textual deconstruction of each question's content would, in turn, facilitate the extraction of topics of discussion related to COVID-19 software development. These topics are expressed through sets of related words revealing the intentions and purposes behind posting a question, without being limited by the tagging labeling mechanism (Beyer et al., 2020). To automatically unveil topics of discussion related to COVID-19, the *Latent Dirichlet Allocation* (LDA) modeling algorithm (Blei et al., 2003) was applied on the corpus of the title and body fields extracted from the set of question posts.

Described briefly, LDA is a popular probabilistic modeling technique utilized for the extraction of topics in a given collection of documents (question posts in the case of our study) and has been widely used in many experimental setups regarding topic extraction in SO (Barua et al., 2014; Villanes et al., 2017; Ahmed and Bagherzadeh, 2018; Venkatesh et al., 2016; Yang et al., 2016). The general idea behind LDA is the representation of documents as distributions of probabilities over a number of latent topics, whereas each topic is represented by a continuous sequence of words that characterize it (Blei et al., 2003). Thus, LDA can be particularly useful in revealing the hidden topics by exploring observable patterns of words that co-occur frequently in a collection of documents. The selection of the number of topics K is a user-defined process and for this reason, the decision-making is totally based on extensive experimentation with different values of K (Barua et al., 2014). Hence, the optimal value is difficult to be defined, since each experimental setup provides meaningful topics with a different value of K (Blei et al., 2003; Ahmed and Bagherzadeh, 2018). Generally, a high value of K facilitates the extraction of deeper, more specific topics, whereas smaller values of K yield more broad and general topics (Papoutsoglou et al., 2020; Barua et al., 2014). In the current study, several experimentations and trial runs of LDA were conducted so as to optimize the value of K. The *Coherence Score* was used as a metric of evaluation for the final selection. The parameter K was finally set to 14, a number of extracted topics that capture, in a satisfactory and coherent way, the content of questions about COVID-19 software development posting activity.

The application of the LDA model on all posts of the corpus returned two data representations that would be later used for analysis. The first was the produced topics, expressed as a distribution $z_k = [(word_1, prob_1), (word_2, prob_2), \ldots, (word_i, prob_i)]$ that covered all the words of the corpus and expressed the probability of each word appearing in a topic. Evidently, words of higher probabilities comprise the general thematic axis of a topic. The second was the topic distribution for each post, expressed as $p_i = (z_1, z_2, \ldots, z_k)$ (Beyer et al., 2020), which contained 14 elements, corresponding to the 14 defined topics and represented the *membership* $\{\theta(p_i, z_k)\}$ of a topic (z_k) (Beyer et al., 2020) in the document p_i in a specific degree, ranging from 0 to 1. For example, in a post with a vector of $[(1, 0.3), (2, 0.05), (3, 0.45), \ldots]$ the first topic presents a 30% membership, the second topic presents a 5% membership etc., with all membership values summing up to 1. We express all topic membership values in percentages in order to facilitate interpretation. It should be noted that our model achieved a *Coherence Score* of 0.6, proving its high efficiency. A common observation from other studies is that coherence values over 0.5 are clear indicators of a well-rounded LDA model (Beyer et al., 2020; Blei et al., 2003).

Based on the LDA model, we evaluated well-known metrics facilitating the interpretation of the extracted results. More specifically, we calculated the *dominant topic* for each post as the topic with the highest membership value (Chakraborty et al., 2021). Formally, the dominant topic of a given post p_i is defined as

$$dominant\,(p_i) = z_k\colon \theta\,(p_i, z_k) = \max\left(\theta\left(p_j, z_j\right)\right);\; 1 < j \le K \tag{5}$$

Concerning the share of a topic, this metric expresses the proportion of posts that contain a specific topic z_k. Following the approach of Beyer et al. (2020), we made use of a threshold δ of 0.1 (or 10%) to remove noisy topic membership values and discard the probabilistic errors. Based on the previous considerations, the share of a topic z_k is defined as

$$share\,(z_k) = \frac{1}{|P|} \sum_{\substack{p_i \in P \\ \theta\,(p_i, z_k) \ge \delta}} \theta\,(p_i, z_k) \tag{6}$$

where $|P|$ is the number of all posts in our dataset.

Finally, in order to trace the collective popularity of dominant topics in the corpus, we calculated the *popularity* metric as (Chakraborty et al., 2021)

$$popularity\,(z_k, P) = \frac{|\{p_i\}|}{|P|}\colon dominant\,(p_i) = z_k;\; 1 \le j \le K \tag{7}$$

where $\{p_i\}$ is the total number of posts that have z_k as their dominant topic and $|P|$ is the number of all posts in the corpus.

While the defined metrics provide a clear insight to the popularity and distribution of topics among posts, they offer limited feedback on the similarity (or distance) between the extracted topics. However, as topic similarity is a very important attribute, proving the robustness and valid formulation of the LDA model (AlSumait et al., 2009; Tong and Zhang, 2016; Celikyilmaz et al., 2010) and tracking shared linguistic and semantic traits between topics, its computation was of high value. Thus, our next objective was to utilize a distance metric that would be suitable for probability distributions. The rationale behind this approach is that each topic z_k, produced by the LD, is essentially a distribution of probabilities among the words of the corpus. Some widely known metrics evaluating the difference between two probability distributions are the Hellinger distance (Rus et al., 2013), being primarily used in Statistical Inference, the Kullback–Leibler divergence (Rus et al., 2013; Somasundaram and Murphy, 2012), being particularly useful in entropy computation of Information Systems and the Jensen–Shannon divergence (Rus et al., 2013), which is a symmetrical version of the Kullback–Leibler divergence. We opted to use the Jensen–Shannon divergence to avoid some problems that the Kullback–Leibler metric creates (non-symmetrical, division with zero) (Niraula et al., 2013). Finally, we also decided to represent the topics distributions in a two-dimensional space to showcase the intertopic distances. For this purpose, we leveraged the PyLDAVis package (Bmabey, 0000) that utilizes *multidimensional scaling* (Cox and Cox, 2008) in order to project the topics in a two-dimensional space facilitating the interpretation of their similarity (or dissimilarity).

5. Results

In this section, we present the findings of this study based on the posed RQs.

[RQ$_{1.1}$] *Did the evolution of COVID-19 pandemic trigger corresponding knowledge-sharing activity in SO and how has this phenomenon evolved over time?*

In order to gain insights about whether the evolution of the COVID-19 pandemic and the associated need for SSD and general software has triggered the initiation of knowledge-sharing activity in SO, in Fig. 4(a), we present the number of daily USA and global confirmed cases using a seven-day rolling average over the examined follow-up period, whereas Fig. 4(b) shows the distributions of questions and answers related to COVID-19. Having a closer inspection of the retrieved Q&A posts, the "post-zero" is traced back on January 26 indicating a swift response to the critical circumstances caused by the outbreak of the health crisis. Indeed, the examination of the distribution of the Q&A posts (Fig. 4(b)) reveals a significantly increasing trend on COVID-19 related posts during the first wave of the pandemic. Moreover, the exploration of the two distributions (number of posted questions versus number of posted answers) demonstrates that SO community has immediately responded to this emergent situation and the rising need for help and advice in software development, a fact that is graphically displayed on the shapes and trends of the two time-series.

Another interesting finding is extracted from the exploration of SO users' activity during the study period. To this regard, there is a notable continuously increasing rate of Q&A posts during the first two months of the pandemic (Fig. 4(b)), when the unprecedented health crisis delivered a global shock to the whole world (Fig. 4(a)). In addition, the rapid activation of SO community may reflect the growing interest on digital initiatives related to SSD for tackling COVID-19 crisis. The overall pattern also indicates a steep rise in knowledge-sharing activity during the first ten days of March 2020, when the number of posts started to steadily decrease by a smoother rate until the end of the first wave of pandemic in the summer.

[RQ$_{1.2}$] *Which are the characteristics of COVID-19 knowledge-sharing activity in SO?*

RQ$_{1.2}$ focuses on meta-characteristics information extracted through descriptive statistics analysis (Table 3) in SO activity and quality metrics (Anderson et al., 2012b) with the aim of understanding the dynamics of COVID-19 software development knowledge-sharing activity. To this regard, except from the findings concerning the final dataset of 2213 question posts related to COVID-19, we also provide the results of the analysis conducted on a dataset comprising general posts that can be used as a reference basis. In particular, we randomly collected 2213 posts excluding posts that were related to COVID-19 using a sliding time window covering a nine-month period. The findings, for the COVID-19 posts show the following:

(a) By the terminal date of the follow-up period, 68% of questions had received at least one answer (Table 3), a percentage that is very close to the overall reported SO performance metric (70%) (Stack exchange, 0000). The percentage of the general posts, in the corresponding time window, that received an answer was 63%. The chi-square test of independence indicated a statistically significant association between the type of post (COVID-19/general) and the receiving of at least one answer, $\chi^2(1) = 12.098$, $p < 0.001$, $\varphi = 0.05$.

(b) Regarding the distribution of the number of received answers of COVID-19 posts, Fig. 5 indicates that 48.89% of questions ($N = 1082$) received strictly one answer, a generally lower percentage than the corresponding reported percentages in other similar studies (Ortega et al., 2014; Neshati, 2017). In comparison, in the general posts this percentage (of posts having only one answer) is quite similar (46.40%). Generally, the Mann–Whitney test revealed a statistically significant difference between the distributions of the two types of post, $Z = 3.501$, $p < 0.001$, $r = 0.053$.

(c) A total of 1931 (87.26%) COVID-19 posts contained code snippets, which is considered a key factor affecting the quality of questions (Treude et al., 2011; Meldrum et al., 2020), since the inclusion of code snippets contributes to the clarification of the issue being asked and thus, it may accelerate the response time of questions (Wang et al., 2018b). In the corresponding general posts, the percentage of code snippets is 82.79% (Table 3). This difference is an indication that COVID-19 posts contain even more specific content. The chi-square test of independence indicated a statistically significant association between the type of post (COVID-19/general) and the indicator variable of contained code snippet, $\chi^2(1) = 20.794$, $p < 0.001$, $\varphi = 0.07$.

(d) Regarding the number of comments of COVID-19 posts that can be used for follow-up by triggering successive rounds of debates about the post by aggregating statements of agreement or disagreement (Anderson et al., 2012b), the distribution (Fig. 5) indicates that 57.70% ($N = 1277$) of posts received at least one comment (Movshovitz-Attias et al., 2013; Diyanati et al., 2020). The corresponding percentage in general posts is 52.15% ($N = 1154$). Finally, the Mann–Whitney test showed that the distributions of question views for COVID-19 and general posts presented a statistically significant difference, $Z = 8.835$, $p < 0.001$, $r = 0.133$.

[RQ$_{2.1(a)}$] *Which technologies are more popular in COVID-19 software development and how are these associated to each other?*

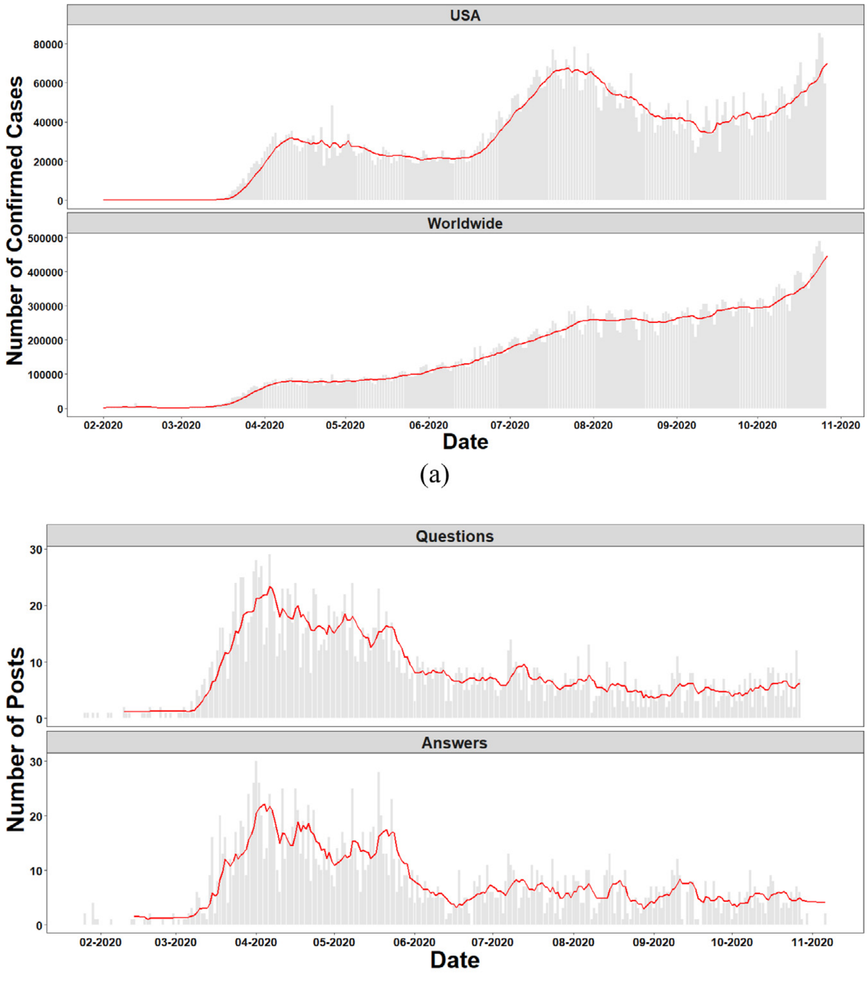
As we have already mentioned, RQ$_{2.1(a)}$ focuses on the identification of broad TCs and prominent technologies belonging to each TC. Based on the categorization of questions into TCs (First Tier of the hierarchy, Fig. 2), 1318 (59.55%) posts were classified into a unique TC, whereas 670 (30.28%) posts were categorized to more than a single TC indicating multifaceted technological issues arisen in COVID-19 software development.

The remaining 225 posts (10.17%) were not categorized to any of the seven broad TCs, since they were not tagged by any of the predefined 182 tags of the reference lexicon. The distribution of question posts indicates that the majority concerns **Languages** related technological problems (64.53%) followed by **Web Frameworks** (11.72%) and **Big Data/ML** (11.69%). In contrast, the broad TCs of **Platforms** (4.81%), **Databases** (2.44%), **Developer Tools** (2.22%) and **Collaboration Tools** (1.11%) accumulate a relative lower percentage of question posts, which is an indicator of generally lower popularity of these specific technology aspects in COVID-19 software development.

At the lowest level of the hierarchy (Second Tier), the exploration of user-defined tags provides straightforward directions about which specific technologies have been mostly adopted by developers in COVID-19 projects. Fig. 6 presents the top recurring technologies for each TC, whereas in Table 4, we indicatively present demonstrative examples of question posts related to each TC in order to facilitate the understanding of specific-technology usage.

To allow comparison with posts lacking a specific theme, the corresponding percentages of TCs within the general posts are: **Languages** (51.2%), **Web Frameworks** (18.2%), **Big Data/ML** (4.7%), **Platforms** (11.1%), **Databases** (7.1%), **Developer Tools** (5.5%) and **Collaboration Tools** (1.8%). Moreover, Fig. 6 contains the distributions of Second-Tier technologies of general posts side-by-side with the corresponding COVID-19 related. The distributions exhibit some similarities: For example, *pandas*, *tensorflow* and *keras* are the most popular libraries for Big Data and Machine Learning in both cases, as it is the case with *reactjs* and *vuejs* for Web Frameworks. However, at a closer look, striking differences can be identified for COVID-19 related posts. *Pandas* is the predominant choice for data analysis which is also in agreement with the much higher presence of *python* (in roughly half of the COVID-19 related posts classified under the Languages TC). Regarding Languages an interesting observation is that *r*, constitutes a highly popular language for COVID-19 related statistical computing/graphics (ranked 2nd) while it is ranked much lower in the case of general posts. A general finding is that data analysis is of utmost importance for COVID-19 related posts. This is further supported by the existence of *c#* among general posts, a language that is out of the scope of data analysis and is not present in the COVID-19 top languages.

Regarding **Languages**, developers appear highly eager in retrieving, processing and analyzing data from different sources, as indicated by the high percentages of the two top programming languages for data science (*python* and *r*). Moreover, increased interest can be observed for visualization and presentation of information as implied by the popularity of tags pointing to frontend technologies (*javascript, html*). Beyond *python* and *r*, *java* also appears as a frequent tag, in line with the overall popularity of *java* among general-purpose programming languages



(a)

(b)

Fig. 4. Distribution of (a) the number of confirmed cases per day in USA and worldwide and (b) the number of questions and answer posts.

Table 3
Characteristics of question posts based on SO metrics.

Variable	Categories	COVID-19 posts		General posts	
		N	%	*N*	%
Answered	*No*	708	32.00	819	37.00
	Yes	1505	68.00	1394	63.00
	Total	**2213**	**100**	**2213**	**100**
Code Snippet	*No*	282	12.74	392	17.71
	Yes	1931	87.26	1821	82.79
	Total	**2213**	**100**	**2213**	**100**
Number of comments	*M (SD)*	1.72 (2.24)		1.62 (2.44)	
	Mdn [min, max]	1 [0,21]		1 [0,21]	
Number of question views	*M (SD)*	86.59 (160.52)		228.05 (4044.9)	
	Mdn [min, max]	50 [7,3653]		41 [3,182293]	

Note: *M*, *SD*, *Mdn*, *min* and *max* represent the mean, standard deviation, median, minimum and maximum values of each examined distribution, respectively.

Web Frameworks express the need for developing application and infrastructures to display information related to confirmed cases, country of interest, deaths and more generally, the dynamics of the pandemic. As expected, specialists rely on dedicated environments to facilitate the development of robust data-driven web solutions. To this regard, *reactjs* and *nodejs* seem to dominate (even more for COVID-19 related posts), since they provide a variety of capabilities for agile Web Development and software engineering, with *vuejs* and *angular* closely following, for the development of web interfaces.

Concerning the exploration of large volumes and complex data (**Big Data/ML** TC), there is a clear dominance of the *pandas* library that provides fast and powerful capabilities for the manipulation and analysis of data structures. In parallel, developers are interested in the adoption of ML advances (*tensorflow*) which in many of the posts are aimed for fitting prediction models on epidemiological data, and particularly, there is a preference for deep neural learning algorithms (*keras*).

Given that the implemented solutions require code development and sharing, developers opt to exploit several **Developer**

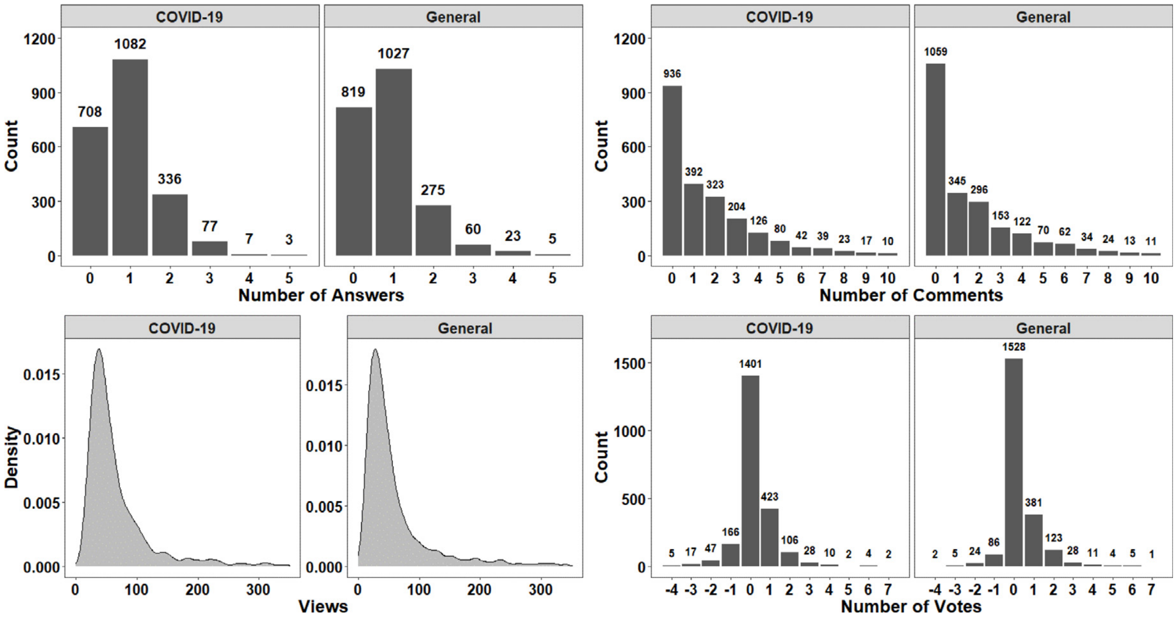


Fig. 5. Graphical representation of distributions for number of answers, comments, views and votes.

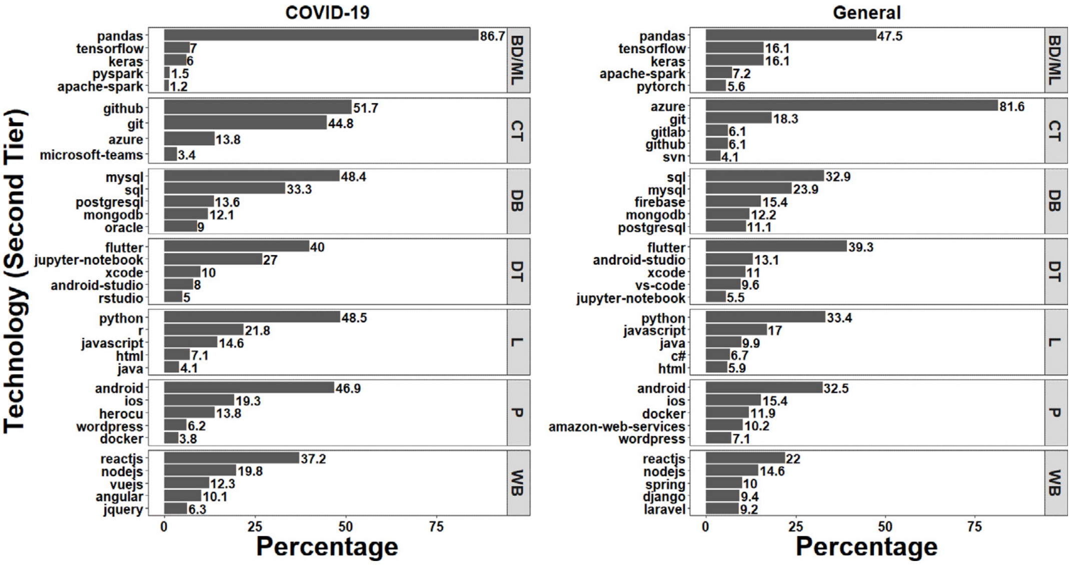


Fig. 6. Distribution of popular technologies for each TC.

and **Collaboration Tools** to facilitate processes and reusability. To that end, self-contained application frameworks (*flutter, android-studio*) and general-purpose code writing suites, particularly for *python* and *r* (*jupyter-notebook, rstudio*) are considered staples in offering solutions related to the pandemic. In addition, *github* retains its position as the most preferred tool for tracking code changes and storing project outcomes in combination with *azure* services for cloud deployment of applications and architectures.

The **Platforms** TC encapsulates full-fledged operating systems and deployment environments, suitable for development and application testing. Given the ubiquitous presence of mobile devices, it is not surprising that the *android* and *ios* operating systems dominate, as specialists swift their attention in delivering high quality applications relevant to COVID-19, such as trackers or applications with infographics (Kelion, 2020). However, Web Development still holds a notable presence, with deployment (*herocu*, *docker*) and development (*wordpress*) suites still being necessary for efficient website creation.

Databases are utilized as a necessary tool in developer needs regarding the COVID-19 pandemic, since they constitute warehouses of valuable data about numbers of confirmed cases, deaths etc. Moreover, applications developed for the support of sectors operating to suppress the pandemic demand well-designed and scalable database schemas that can cope with streaming data. *mysql* is the most prominent technology, being a well-documented for database development, along with *sql* for querying and data retrieval. *postgresql* and *mongodb* are also presented in noteworthy percentages, them being indispensable tools for geographical coordinates and text data storage.

As far as the investigation of the interconnections between adopted technologies is concerned, we graphically explored the associations between the set of seven broad TCs and a subset of specific technologies from the Second Tier of the hierarchy (corresponding to **Languages**, **Web Frameworks** and **Big Data/ML**). The latter choice was due to the fact that the number of question

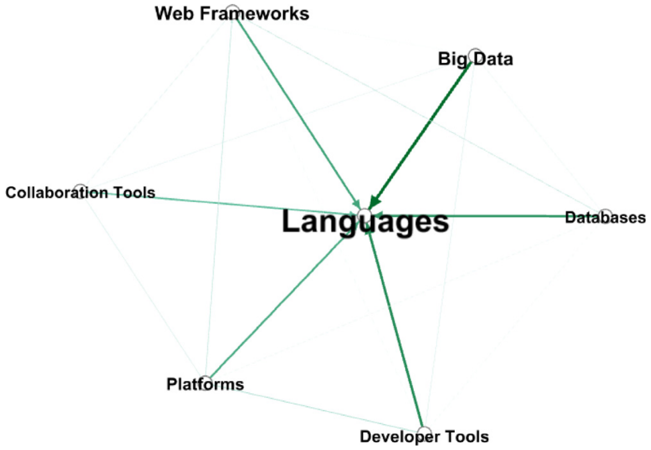


Fig. 7. Association Rule Graph for TCs (First Tier).

posts for the remaining TCs is significantly lower and thus, they do not constitute a strong basis for the construction of ARGs.

Fig. 7 visualizes the ARG interconnections between the broad TCs of the First Level. Generally, the font size is proportional to the relative frequency of the tag in each TC, whereas the thickness and direction of the edges are directly dependent on the *conf* $\left(tag_i, tag_j\right)$ metric. A close inspection reveals that the **Languages** node is quite central, with every other node having an edge directed towards it. In addition, the connections with the **Web Frameworks** and **Big Data** nodes are stronger as indicated by the edge weight. This means that posts referring to these particular two TCs most probably will contain tags or references to the **Languages** TC as well. Similar connections are observed with the rest of the TCs (**Collaboration Tools**, **Developer Tools**, **Databases**). This is an anticipated result, as the **Languages** TC encapsulates many useful technologies that facilitate software related projects in all other areas. Regarding other TCs, **Web Frameworks** and **Big Data** also have several incoming edges, **s**howcasing that these two TCs are commonly referenced in posts. Finally, TCs such as **Databases** and **Collaboration Tools** have no incoming edges, acting as supplementary factors that reference more established TCs in related posts.

The ARG constructed by tags found in question posts that belong to the **Languages** TC (Fig. 8) indicates the prominence of *python* and *r*. As proven by the examination of the inter-TC frequencies, these two programming languages are widely used in COVID-19 software ventures for the development of epidemic and Machine Learning models as well as the analysis and processing of data. Their robustness and maturity in Data Science practices is validated by their interconnections with other technologies, including tools for web scraping (*beautifoulsoup*, *selenium*), graphical visualization (*matplotlib*, *plotly*, *seaborn*, *choropleth*) and geospatial models (*geopandas*). These associations reveal the nature of COVID-19 projects such as the design of visualization simulations for new cases and deaths, the extraction of information from multiple sources and the geospatial tracking of the pandemic's spread. Moreover, *python* is directly linked with the **Big Data** TC via the *pandas* tag, indicating the indispensable connection between these two technologies and the desire for scientific data wrangling. Finally, a connection with typical *python* structures (*dataframe*) is observed, further confirming the status of *python* as a reliable **Language** for Data Analysis. The *r* tag also presents noteworthy interconnections, even more so than *python* concerning the design of statistical models, as it is connected with classic mathematical and machine learning terms (*regression, time series)* and the necessary web-scraping packages (*rvest*). Complementary to the design of models is the graphical representation of the results, as shown by the presence of a well-known visualization package (*ggplot2*) and visualization tools (*leaflet*). An intuitive interpretation of these connections is that developers strive to understand and simulate spreading patterns in combination with proper visualization, with a possible intent of conducting an epidemic analysis. Finally, an interesting cluster is observed for *javascript*, concerning the development and deployment of websites and applications. Connections with relevant **Web Frameworks** (*reactjs, vuejs, nodejs, angular)* reveals the increased activity in creating tools and software products that contain relevant information regarding COVID-19.

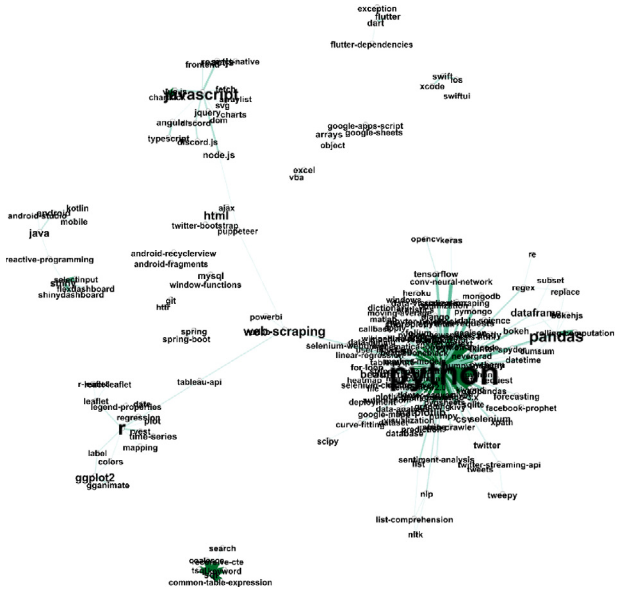


Fig. 8. Association Rule Graph for **Languages** (Second Tier).

An initial inspection in the **Web Frameworks** ARG (Fig. 9) further indicates the main frameworks that are widely used by the community for the construction of websites or mobile applications. *javascript* and *reactjs* remain the prominent terms, with *nodejs* closely following, showcasing the increased preference for self-contained tools that facilitate the design and implementation of a website. However, *python*-based frameworks (*django, flask*) are also used, though in a smaller percentage. In addition, close connections with the **Databases** (*mongobb, mysql, postgresql*) and **Platforms** (*herocu, docker*) TCs are observed, referring to Full Stack Development and deployment of COVID-19 related tools. The **Languages** TC is also present (*html, css, python*) as some core languages are required for development while an interesting finding is that other general terms (*web scraping, data visualization*) refer to different aspects of web elements manipulation that are also crucial for COVID-19 software development.

The **Big Data/ML** ARG (Fig. 10), though containing fewer nodes and edges still includes some important technologies and interconnections. As expected, the most prominent **Big Data** tag (*pandas*) is directly connected with the most frequent term of the **Languages** TC (*python*), a finding that has already been mentioned and proves the dependency of Big Data analytics on appropriate programming languages. Furthermore, apart from a separate cluster dedicated to ML and deep neural network algorithms, which contains key relevant terms (*tensorflow, keras, lstm, statistics*), a separate cluster relevant to scalable Big Data Analysis is observed with direct reference to *apache-spark* and *pyspark*. The need for data storage, especially in large volumes is also expressed by the presence of **Databases** tags (*mysql*), even though

Table 4
Indicative examples of question posts for each TC.

TC	Example of question post
Languages	*I am trying to generate predictions for covid cases using a GAM model. The following code works and produces a projection of US cases. I want to apply the same code to any country I choose and therefore thought a simple function would be easiest. This function basically takes all the code above and wraps it up in a function. However — initially it failed at the data subsetting line, so I put the get_df as a helper function. Now it fails at the gam analysis. The error seems to suggest that data$day_num does not equal data$ location length, but I can't work out why it would say that because they are the same length. I've read the various responses on stack overflow and can't find any answers and hunting around the internet has turned up anything either. I'd greatly appreciate any help! To get the covid data for a fully reproducible example:*Tags: **r** data-science data-modeling gam
Web Frameworks	*I am trying to create a simple app with vue-cli and the router that fetches Covid-19 cases by Country from a JSON object of arrays. This is my first Vue app. However, I keep getting an error about "Declaring Reactive Properties". I searched dozens of similar errors on many different forums and seemed to do the trick. Most of the code is from vue.org, except for the JSON link.Api.js:.About.js...Error:.[Vue warn]: Property or method "errored" is not defined on the instance but referenced during render. Make sure that this property is reactive, either in the data option, or for class-based components, by initializing the property. I can see the warning 3 times, for each of the props errored, loading and info, the most important one.*Tags: **vue.js** vue-router
Big Data/ML	*I am a beginner with TensorFlow and neural networks. I need to run ANN between Protein Sequence and Its Target Molecules. I want to use this trained network for prediction of drug molecule for nCoV-2019. I am struggling with codes here. I get error: Failed to convert a NumPy array to a Tensor (Unsupported object type float).Please help me to write appropriate codes here. Thank you in advance.*Tags: **tensorflow**
Developer Tools	*I am building a project about coronavirus tracker for my class. Everytime I tried to call the api and hit the search icon it throws an error in the console. I have tried many things but it throws the same error. can anyone help figure out what is wrong with code? Here is the Api code. Here is the constructor*Tags: **flutter** flutter-layout flutter-dependencies
Collaboration Tools	*I have created a repo in GitHub named Covid-19-Predictor-BD. I have also linked it up to GitHub Pages which you can see at https://abd-shoumik.github.io/Covid-19-Predictor-BD/ But when I search 'Covid-19-Predictor-BD' or some related keyword in google , my repo doesn't appear in the search. What I can do to make the repo appear in google search?*Tags: **github** google-search
Platforms	*I'm making a Pi livestreaming covid tracker but watching my CPU/RAM get decimated. When I run that, top gives me %CPU 282%MEM 4.0.I installed ffmpeg with sudo apt-get ffmpeg. I'm not sure if I'm using hardware acceleration. It was brought up in a number of posts that were a year old or so.*Tags: ffmpeg **raspberry-pi** video-streaming
Databases	*I'm building an Api that retrieves time series data of covid cases for each country in NodeJS/Express with a mongo DB. The data source is courtesy of John Hopkins. For those lazy to hit the link, the headers look like this: A typical row: I'm trying to model the .csv into a schema that allows for CRUD operations. Having trouble wrapping my head around how each date will dynamically be added on a daily basis. Currently, I have: How best should do I design the mongo schema based on the .csv? Thanks!*Tags: **mongodb** mongoose mongoose-schema

the absence of other connections could indicate that developers may refer to other forms of storing for data files (e.g. Pickle, Excel, CSV files etc.). In addition, nodes directed towards *pandas* further reveal **Big Data** activities related to COVID-19, ranging from time series manipulation (*time series*) to geographical mapping (*geopandas*) and plotting (*matplotlib*). A notable finding is the connection of an HTTP requests software (*axios*) directly with *api* and **Web Frameworks**/**Platforms** terms (*android, reactjs, nodejs*), which may refer to the simultaneous retrieval of large volumes of data via the use of integrated tools and the immediate plotting in applications or websites. Finally, web scraping remains a heavily practiced activity, with relevant terms (*beautifoulsoup, selenium*) occurring in all ARGs.

[**RQ$_{2.1(b)}$**] *Which technologies present more difficulties in COVID-19 software development?*

Having identified prominent technologies and interconnections, the next step involves the investigation of whether specific TCs raise higher levels of difficulty in knowledge-sharing activity than others. At this point, we have to clarify again that a question post may be categorized into more than a single TC. Table 5 summarizes the distributions of question posts for each TC that received at least one answer during the examined period. The findings indicate generally high percentages (above 70%) for the majority of TCs but they also unveil a specific TC (**Platforms**) presenting significantly lower percentage compared to other TCs with a high number of unanswered (censored in Survival Analysis terminology) posts.

More importantly, the examination of the duration distributions for time elapsed until the first answer through K–M curves (Fig. 11) provides noteworthy findings related to the difficulty of question posts according to the class that are belong to. The early steep decent of curves representing the distributions of **Big Data**, **Web Frameworks**, **Languages**, and **Databases** related posts indicates that these are more likely to receive prompt feedback compared to question posts belonging to other TCs. In contrast, the flat shape with long horizontal gaps for **Platforms** and **Collaboration Tools** distributions shows that it takes significantly longer time for these posts to receive a first answer, which may practically indicate the need for higher level of expertise. Indeed, the evaluation of the median values for each TC depicts notable divergences for response times (Table 5). At this point, we also have to note that our preference to the more robust central tendency measure of median rather than the mean value, is due to the highly skewed duration distributions. The comparison of median values, which are also graphically presented on K–M curves (dashed lines), demonstrates that both **Platforms** ($Mdn = 88.25$ hours) and **Collaboration Tools** ($Mdn = 33.24$ hours) questions present very long response times, whereas **Big Data** ($Mdn = 0.95$ hours), **Web Frameworks** ($Mdn = 1.33$ hours), **Languages** ($Mdn = 1.99$ hours), **Databases** ($Mdn = 2.44$ hours) and **Developer Tools** ($Mdn = 4.96$ hours) are resolved through a more responsive way by the SO community. Finally, the *log-rank test* indicated statistically significant differences between

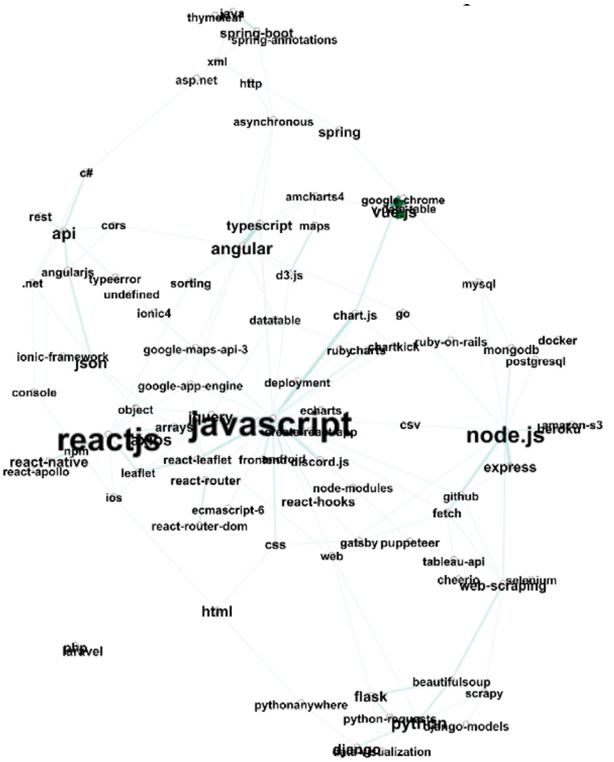


Fig. 9. Association Rule Graph for **Web Frameworks** (Second Tier).

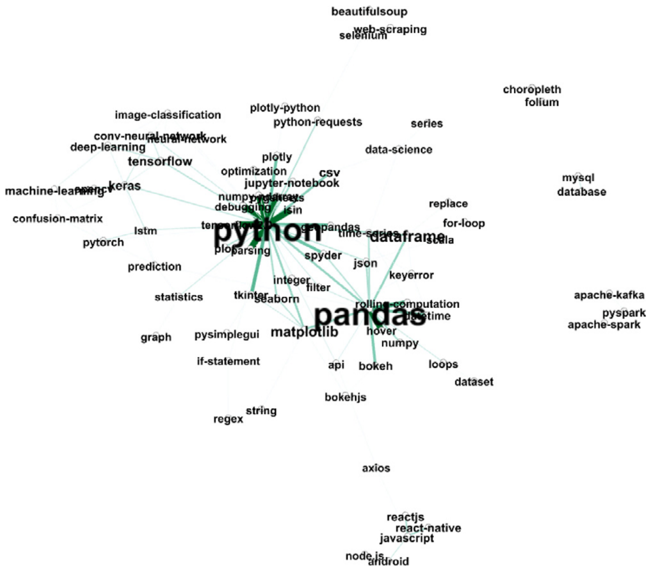


Fig. 10. Association Rule Graph for **Big Data/ML** (Second Tier).

the median values of time elapsed until the first answer (χ^2 (6) = 32.7, $p < 0.001$).

Regarding the percentage of answered questions for general posts and the median time to first answer (Table 5), a first remark concerns the percentages of unanswered questions that are higher compared to COVID-19 posts for all except one TC (**Collaboration Tools**), where values almost identical. Moreover, an interesting finding is the fact that the median response times for general posts are significantly higher for five (**Big Data, Web Frameworks, Languages, Developer Tools, Platforms**) out of seven TCs. The median response times are quite close for **Languages**, whereas there is noted a higher median response time in COVID-19 posts for the **Collaboration Tools** TC compared to general posts.

[RQ$_{2.2}$] *Which topics are more popular in COVID-19 software development?*

In contrast to RQ$_{2.1(a)}$ that is more related to technological aspects of the question posts, RQ$_{2.2}$ aimed to exploit the wealth of textual information hidden in the *title* and *body* features. The semantic insights would provide us with directions regarding the prominent topics of discussion in COVID-19 related posts. This statement is strengthened by the fact that, while *tags* are a very brief and efficient representation of the technological orientation of a question, the true intent and motives that prompted a user to engage in conversation on the SO ecosystem will inevitably be discovered through the analysis of its textual content.

Table 6 summarizes the results of LDA extracted by setting the parameter value of K equal to 14. In addition, to the best of our ability, we manually assigned a short description to each one of the extracted topics based on the sets of their associated key words in order to better illustrate the general purpose of question posts. An early inspection of the produced topics reveals that, though some of them refer to scientific inquiries related to the COVID-19 pandemic (e.g. Topic 3, Topic 14), a notable portion is relevant to generic tasks such as app & web development (Topic 11) or the retrieval of elements from web sources (Topic 5). This result is more than anticipated, because while the software solutions related to combatting the pandemic would be relevant to exclusive traits such as deaths, reported cases and infections, the specialists developing these solutions would still need guidance for common problems that the software community faces.

A common trait observed in most topics is that, while the majority of them concern different or partially connected aspects of COVID-19 software development, their nature is inquisitive. This can be attributed to the presence of words such as "*how*", "*help*" and "*try*" that can be related to enthusiasts or individuals that actively engage in projects concerning COVID-19 and seek solutions to their problems. Moreover, the fact that the majority of topics refer to practical software related projects further indicates the involvement of specialists from various domains and backgrounds in developing software that can contribute to the battle against COVID-19.

In addition, Table 6 summarizes the share metric values, indicating that Topic 1 and Topic 11 are the two most shared topics across all question posts. These two specific topics of discussion concern the retrieval and storage of data related to COVID-19 pandemic via integrated APIs from different web sources and the development of applications that dynamically update and present relevant information. Given that data retrieval is essential for any developed solution related to the pandemic and is related to the construction of web applications for displaying information, their large share values are not surprising. In contrast, Topic 13 (Geostatistics) presents a significantly lower share value (26.02%). This can be possibly attributed to its more technical nature that requires practical and specific knowledge, barring it from being shared in a large number of posts.

In contrast to the share metric, that takes into account the whole distribution of membership values for a given topic, the popularity metric can be used for the identification of dominant topics across the collection of question posts. To this regard, the retrieval and storage of data via APIs (Topic 1) seems to be the most dominant topic with a popularity value of 18.4%. The second most popular topic of discussion (Topic 2) concerns the daily monitoring of COVID-19 cases across different countries. The increased popularity of these topics showcases the primary objectives of the development of COVID-19 related solutions, which are the manipulation of time series expressing the evolution of cases in a local or global scale as well as the tracing

Table 5
Distribution of answered questions and median time elapsed until the first answer.

Technology class	COVID-19 posts			General posts		
	N	*#Answered* (%)	*Mdn*	*N*	*#Answered* (%)	*Mdn*
Big data	316	245 (77.53)	0.95	124	80 (64.52)	6.37
Web frameworks	317	219 (69.30)	1.33	477	301 (63.10)	19.16
Languages	1745	1222 (70.03)	1.99	1339	892 (66.61)	5.53
Databases	66	48 (72.73)	2.44	188	129 (68.62)	2.53
Developer tools	100	68 (68.00)	4.96	145	81 (55.86)	124.75
Collaboration tools	30	22 (73.33)	33.24	49	36 (73.47)	21.49
Platforms	130	74 (56.92)	88.25	292	157 (53.76)	308.55

Note: *Mdn* represents the median time elapsed (in hours).

and retrieval of relevant data. The least popular topics are the application of Geostatistics to trace geographical patterns (Topic 13) and potential issues during the creation of charts (Topic 12), though the decreased popularity of the latter can be explained by posts that express this need and may be matched with other Topics (e.g. Topic 4, Topic 8).

Regarding the interrelations between the extracted topics, Fig. 12(a) presents the pairwise distances between topic distributions based on LDA using the Jensen–Shannon divergence, whereas Fig. 12(b) visualizes the projection of distances on a two-dimensional space via multidimensional scaling. In this figure, the area of the circle is proportional to the prevalence of the topics in the corpus, whereas the centers of the circles are positioned according to their intertopic distances. Generally, a meaningful LDA model should be represented by large-sized and segregated circles (Sievert and Shirley, 2014). To this regard, the projection of the extracted topics on the two-dimensional space indicates a meaningful LDA solution, since the majority of the circles are non-overlapping and are present in all the quadrants of the plot. Topic 1, related to the retrieval of information via APIs, is the dominant topic while Topic 13, related to Geostatistics, is the least discussed topic.

A first interesting finding concerns Topic 13 (Geostatistics) represented by a circle that is positioned significantly far away from the bulk indicating a topic of discussion that is generally dissimilar to the rest. The other topics present varied values of dissimilarity, with Topic 3 (COVID-19 Data Visualization) having the closest distance score with Topic 1 (Retrieval and Storage of Information via APIs), a rational finding as the objective of Topic 3 is heavily dependent on the retrieval and storage of necessary elements expressed by Topic 1. In general, as many topics are interweaved, and concern generic problems, a degree of similarity is more than anticipated. For example, the objective of Topic 2 (Time series analysis) can be a part of COVID-19 visualization, as expressed by Topic 3, while the plotting of data (Topic 6) requires meticulous data extraction from web sources (Topic 5). However, the distance scores showcase that each topic can still be interpreted as separate from the other topics. Indeed, many topics seem separate from the central core of circles and seem to focus on different areas, such as data extraction from articles (Topic 7), error handling (Topic 8, Topic 9, Topic 10) and app development (Topic 11).

6. Discussion and implications

In this section, we review major findings of this study and provide our own interpretation on the reasons behind them or potential implications to researchers and practitioners.

The results from $RQ_{1.1}$ revealed that knowledge-sharing communities such as SO respond fast to emerging issues and crises proving their timeliness and keen interest of participants to contribute to an open exchange of ideas and solutions. Moreover, it appears that the interest of developers on COVID-19 related issues and challenges was sustained after its initial peak, possibly pointing to a promising further exploitation of open medical data in the future. The post meta-characteristics ($RQ_{1.2}$) indicate that software development on COVID-19 is a relatively young field: despite the fact that many of the programming questions have been answered in other domains (i.e. how to collect, store and visualize data), the corresponding knowledge might be less spread among researchers and developers working on COVID-19. While the percentage of questions receiving an answer is comparable to the overall reported SO performance metric (∼70%), almost 72% of questions received a single answer.

The analysis of tags in SO posts ($RQ_{2.1(a)}$) reveals, with a high degree of certainty, that the problems addressed by software

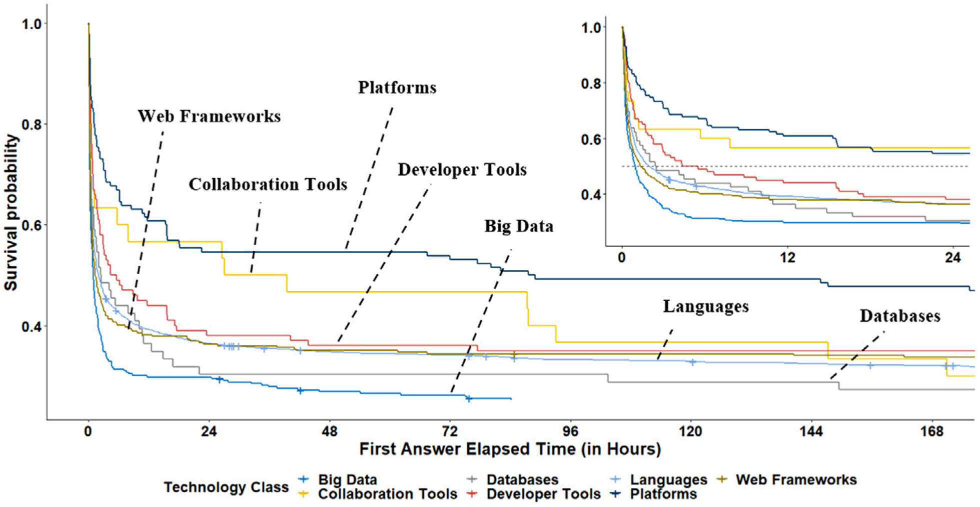


Fig. 11. Kaplan–Meier curves for time elapsed until the first answer for each TC (COVID-19 posts).

Table 6
Topics extracted by the LDA method.

Topic interpretation	Key words	Share%	Popularity%
Topic 1 (*Retrieval and storage of information via APIs*): Usage of integrated APIs in order to accelerate the collection process and provide rapid analytics.	*data, use, try, api, get, code, how, column, file*	62.73	18.4
Topic 2 (*Time series analysis*): Analysis and plotting of time series presumably for monitoring the evolution of cases or deaths in a specific timeframe.	*case, data, date, try, how, day, country, use, column, number*	50.89	14.2
Topic 3 (*COVID-19 data visualization*): Modeling in maps or in graphs of figures concerning the reported cases, presumably to study the epidemical spreads and examine the pace of transmission.	*use, data, code, how, try, map, case, work, plot*	57.84	9.7
Topic 4 (*Support for code related data analysis*): Questions that express the need for other individuals to review a data analysis code snippet and provide suggestions or feedback.	*use, value, how, get, data, code, want, try*	51.37	6.7
Topic 5 (*Data extraction from web elements*): Exploitation of web sources in order to retrieve COVID-19 related data found in several web elements.	*data, file, try, button, use, download, how, get, code, click, url*	50.95	7.1
Topic 6 (*Data plotting*): Plotting specific data related to COVID-19 (cases, deaths etc.) in multiple ways.	*use, data, get, try, how, select, like, want, plot*	58.68	6.9
Topic 7 (*Text retrieval from online sources (e.g. articles)*): Text mining and text scraping from sources that provide detailed reports about the latest developments around COVID-19.	*use, code, how, get, article, try, word, list, what, text*	54.88	5.4
Topic 8 (*Error Handling / Suggestions in website design*): Solutions in technical errors and bugs that occur during the development phase of COVID-19 related websites.	*try, data, get, code, use, how, error, work, website, want*	58.30	5.9
Topic 9 (*Suggestions for code enhancements*): Suggestions of enhancements that can make code snippets run more efficiently and smartly.	*use, data, file, code, work, try, how, would, number*	55.42	5.1
Topic 10 (*Technical issues on data usage and file creation*): Potential obstacles that a specialist could be subjected to when using data or storing information in files.	*use, try, data, code, get, file, error, how, work, create*	58.16	5.7
Topic 11 (*App & Web development / Data updating*): Creation of informative mobile applications and websites that may operate as trackers of cases, update data regarding the demographics of the pandemic (deaths, cases etc.).	*data, use, try, how, value, get, work, help, app, update*	62.12	5.4
Topic 12 (*Issues in creating charts*): Questions in this topic seek solutions in technical issues regarding chart visualizations.	*get, code, line, bar, try, file, chart, how, use*	45.69	3.8
Topic 13 (*Geostatistics*): Plotting of several types of features in interactive maps / Exploration of geographical patterns concerning the number of confirmed cases, deaths or recovered patients.	*type, latitude, longitude, feature, coordinates, property, point, median, state, zip code*	26.02	1.6
Topic 14 (*Data for cases/ deaths per country*): Issues encountered during the analysis of data concerning cases or deaths distributed for every country.	*get, try, use, data, how, death, country, error, show, case*	57.16	4.1

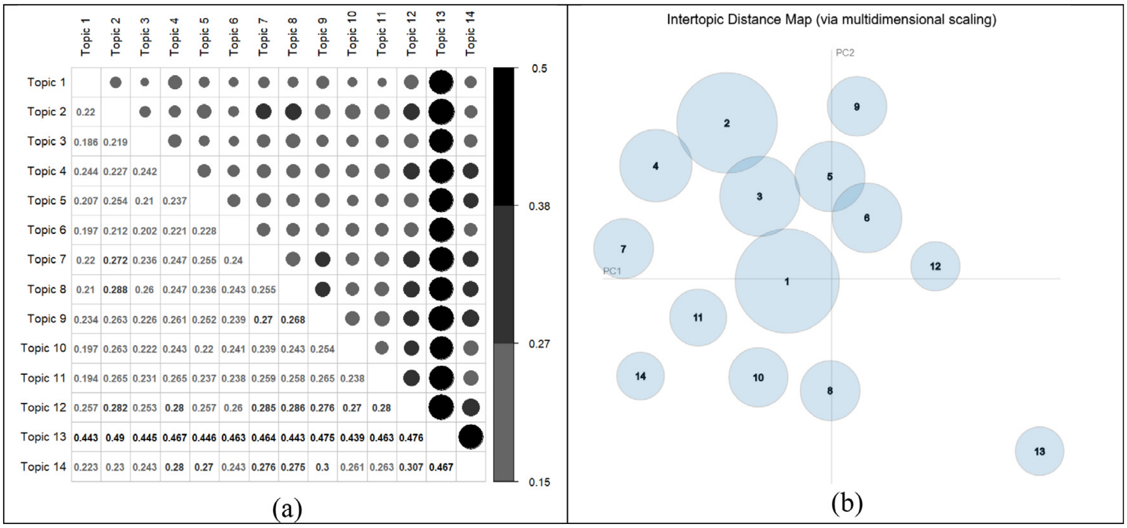


Fig. 12. Intertopic distance (a) scores (b) map.

development specialists and enthusiasts related to COVID-19 pertain to data collection, analysis and visualization. The majority of tags point to languages such as *python*, *r*, *javascript* and *html* and frameworks such as *reactjs*, *vuejs* and *angular*, which are largely targeted at this kind of software development. Furthermore, the investigation of the interconnections among the adopted technologies revealed that the most popular broad topic of **Languages** is often a concern for developers seeking help on problems related to other topics such as **Big Data**, **Databases**, **Developer** and **Collaboration Tools** and **Web Frameworks**.

The responsiveness of the SO community to questions pertaining to COVID-19 questions is quite impressive: based on the results of $RQ_{2.1(b)}$ the median elapsed time for a question to get an answer is less than two hours for the three most popular technology classes, revealing an active and eager to collaborate community of researchers and developers. The comparison with general posts revealed that SO users are more eager to reply COVID-19 related posts in a short time. This could possibly be attributed to the increased interest on issues related to the pandemic itself or because the corresponding COVID-19 related questions refer to already solved problems in other domains.

The analysis of post topics by applying LDA modeling on the titles and bodies of question posts (($RQ_{2.2}$), also revealed that, among others, researchers and developers are highly interested in tracking the COVID-19 phenomenon. The extracted topics include the representation of geographical information, plotting of information over time, retrieving data from online sources, etc. Whether the purpose of the developed software was to simply post information on web pages, provide a comprehensive source of data to other researchers or to systematically delve into the epidemiologic characteristics remains to be studied. Nevertheless, it demonstrates the potential of networked communities of open-source software developers.

In terms of implications to practitioners and researchers, the present study contributes to a better understanding of the strengths and limitations of knowledge-sharing communities such as SO. The breadth of available information, the high responsiveness, and the wide topic coverage prove that SO can form a reliable source of information, at least for newcomers seeking solutions to problems, when they lack the time to perform a thorough training on the involved subjects. While fragmented learning has been criticized for leading to lack of comprehensiveness and systematic thinking, one should acknowledge that for rapidly advancing technological fields, and especially under time pressure as in the case of pandemics, the convenience of Q&A forms offers the benefit of time efficiency. Further indices, beyond ratings and popularity, could be investigated so as to direct developers to the most reliable sources of information, while also considering the criticality of properly analyzing and presenting sensitive, health-related data.

The knowledge obtained from the findings of this study can certainly provide guidelines to data scientists and practitioners, aiding them to focus their attention on the key tools and technologies necessary for the development of scientific software. The results on $RQ_{2.1(a)}$ and $RQ_{2.2}$ reveal that Python is the language of choice when it comes to data analysis and manipulation coupled with libraries like Pandas. The TensorFlow open source library is highly popular for developing and training Machine Learning models while Keras is the first choice for Deep Learning models. When faced with the task of creating functional web tools, developers show a clear preference to the reactJS library (followed by vue.js and angular) for creating views and interactive user interfaces mostly embedded in single page applications. On the other hand, node.js appears to be the most frequently used environment for server-side programming producing dynamic web page content. The next most popular server-side technologies are Spring, Django and Laravel depending on the used programming languages (Java/Python/PHP). Thus, data enthusiasts and software specialists should emphasize on honing these popular digital skills for further strengthening their grasp on developing scientific software solutions.

With respect to software developers and researchers one can observe that while great progress has been achieved through scripting languages (such as *python*), powerful and easy-to-use libraries (such as *pandas*) and interactive environments (such as *jupyter*), emphasizing on code readability, the set of repeating questions arising in Q&A forums implies that there is still room for improvement. One interesting research direction would be the integration of knowledge-sharing channels within the tools employed by the developers and the exploitation of machine learning for recommendation, even without explicitly asking the questions. We also consider the analysis of user's characteristics very interesting in order to shed light into the communities of developers and researchers behind SO posts, their particular interests and problems and also the practices followed for software development in each community.

In any case, the authors consider as a very positive and promising sign the fact that developers in knowledge-sharing communities are eager to collaborate and help others in the face of global challenges.

7. Threats to validity

In this section, we analyze and discuss potential threats to the validity of the present study. Regarding the internal validity, the identification and retrieval of posts relevant to Covid-19 pandemic topics was conducted by an automated process by leveraging the search engine of SO. Though the search strings were broad, we ensured that they were capturing the spectrum of the pandemic, since we included general terms related to coronavirus, to identify a high amount of relevant posts. However, there is always the risk of questions being omitted, where a different terminology or characterization for the pandemic might have been used. For example, there might be posts from users interested in developing COVID-19 related software without explicitly revealing their intentions in the post text, leading to potential false negatives. Nevertheless, we believe that this case does not represent the typical scenario, since the majority of users usually provide a short description of their goals into the body of the post. In addition, we discarded non-relevant posts, which might have contained the keywords of the search string but were expressing a situation associated to the general consequences resulting from the coronavirus lockdown (e.g. issues related to the exploitation of collaborative technologies or remote working). To tackle and reduce to the bare minimum the bias from this process, data filtering was performed independently but simultaneously by the first and the second author and potential conflicts of judgment were discussed and resolved. Moreover, the collection, pre-processing and analysis of data were conducted with the aid of mature packages of *python* and *r*.

Moving on to encountered obstacles during the analysis of data, the extraction of topics by the LDA algorithm proved to be a challenging task, as the proper number of topics used by this method is frequently up for debate. To that end, various experimental setups of the algorithm for all questions were constructed with adjustments for the parameters and the number of topics. The final selection was achieved while considering the general scope of this research and the corresponding research question, which is to track topics of discussion in COVID-19 related posts that reflect the state of software development over the course of the pandemic. Since LDA simply detects latent concepts contained in the procured corpus, the study of the extracted topics can create possible misconceptions, as manual interpretation is inevitable and should be cross validated by the experts of the scientific domains reflected in the topics. In the current study, we resolve possible miscalculations in topic extraction by meticulously examining the topics produced for all experimentations and selecting the setup that optimally reflects the latent concepts of the posts.

With respect to $RQ_{2.1(b)}$, and in order to investigate the level of difficulty faced by specific Technology Classes (TC) in COVID-19 software development we relied on the time elapsed for a post to receive its first answer. Using the time-to-response as a proxy of difficulty entails a construct validity threat, in the sense that beyond the inherent difficulty of the question, the elapsed time is also related to the availability of experts in a field. This threat is partially mitigated by the fact that most COVID-19 related questions deal with generic topics such as data collection, analysis and visualization on which numerous expert users are active.

As for the external validity of the current research, we deem as a notable limitation the application of our methodological framework and the subsequent inferential stages exclusively on posts in SO. While this initial implementation on this particular community can be justified, as SO holds a respected position and popularity in the preferred Q&A sites of software developers, ample opportunities for extended research for comparison and generalization of the findings are offered in other knowledge-sharing forums. Moreover, as we are investigating the impact of a particular and quite recent phenomenon on software development, the timeframe of study was inevitably restricted, and the collection phase retrieved a specific number of posts, undoubtedly smaller in comparison to other studies who explore more general and established topics. However, the importance of our study overcomes the time restrictions, given that the COVID-19 pandemic was an emerging and unanticipated situation with catalytic consequences to all facets of human activity. For this reason, the severity of the situation demands the development of different and necessary strategies and initiatives to comprehend and mitigate its impact, even in this premature form. Finally, the topic extraction analysis was conducted only on the questions of posts. While it is expected that any indications regarding the problem being addressed in the post will be provided in the question, the inclusion of the corpus found in the answers may improve the produced topics, despite the potential introduction of noise.

8. Conclusions

The COVID-19 pandemic will be remembered as a turning-point because of its tremendous impact on the health of millions of people. At the same time, the willingness of the global research community to collaborate for fighting a common battle should be regarded as a very encouraging sign. Knowledge-sharing communities, such as Stack Overflow, underline this attitude of collaboration, since developers around the world exchanged information on how to collect, analyze, visualize and store data pertaining to the pandemic. In this study, we have attempted to investigate COVID-19 related activity reflected on Stack Overflow posts.

The results on the evolution of posts revealed that the response of the developers' community was immediately triggered once the pandemic was declared and has been sustained throughout the crisis. Developers are mostly interested on technologies allowing the collection and posting of data from/to the web, the organization and storing of information, and the visualization and presentation of COVID-19 facts in various forms such as maps and charts. Dominant technologies include Python, R and Javascript while key areas of posts refer to Languages, Web Frameworks and Big Data/Machine Learning. The COVID-19 related software developer community is probably a novel and less mature one: this is hinted by the relatively low number of answers, the occurrence of questions which have been answered in other domains and the longer time to provide an answer for certain topics. Nevertheless, we posit that knowledge-sharing communities can be extremely valuable even to software developers originating from other domains and can strengthen the collaboration towards common goals.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Ahasanuzzaman, M., Asaduzzaman, M., Roy, C.K., Schneider, K.A., 2016. Mining duplicate questions of stack overflow. In: 2016 IEEE/ACM 13th Working Conference on Mining Software Repositories (MSR). IEEE, pp. 402–412.

Ahmed, S., Bagherzadeh, M., 2018. What do concurrency developers ask about? a large-scale study using stack overflow. In: Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement pp. 1–10.

Allamanis, M., Sutton, C., 2013. Why, when, and what: analyzing stack overflow questions by topic, type, and code. In: 2013 10th Working Conference on Mining Software Repositories (MSR). IEEE, pp. 53–56.

AlSumait, L., Barbará, D., Gentle, J., Domeniconi, C., 2009. Topic significance ranking of LDA generative models. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, Berlin, Heidelberg, pp. 67–82.

Anderson, A., Huttenlocher, D., Kleinberg, J., Leskovec, J., 2012. Discovering value from community activity on focused question answering sites: a case study of stack overflow. In: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 850–858.

Anon, 0000a. Build software better together. Retrieved April 10, 2021, from https://github.com/search?q=covid.

Anon, 0000b. About digital response To covid-19, Retrieved April 10, 2021, from https://joinup.ec.europa.eu/collection/digital-response-covid-19/about.

Anon, 0000d. Coronavirus disease (COVID-19. Retrieved April 10, 2021, from https://www.who.int/emergencies/diseases/novel-coronavirus-2019).

Anon, 0000c. Open-access data and computational resources to address covid-19. Retrieved April 10, 2021, from https://datascience.nih.gov/covid-19-open-access-resources.

Anon, 0000. Retrieved April 10, 2021, from https://www.nltk.org,

Anon, 2020. Usdatagov@usdatagov, & usdatagov. Data.gov. Retrieved April 10, 2021, from https://www.data.gov/.

Anon, 2021a. Data on country response measures to covid-19. March 25. Retrieved April 10, 2021, from https://www.ecdc.europa.eu/en/publications-data/download-data-response-measures-covid-19.

Anon, 2021b. Manifesto for Eu COVID-19 RESEARCH. Retrieved April 10, 2021, from https://ec.europa.eu/info/research-and-innovation/research-area/.

Arvanitou, E.M., Ampatzoglou, A., Chatzigeorgiou, A., Carver, J.C., 2020. Software engineering practices for scientific software development: A systematic mapping study. J. Syst. Softw. 110848.

Asaduzzaman, M., Mashiyat, A.S., Roy, C.K., Schneider, K.A., 2013. Answering questions about unanswered questions of stack overflow. In: 2013 10th Working Conference on Mining Software Repositories (MSR). IEEE, pp. 97–100.

Barua, A., Thomas, S.W., Hassan, A.E., 2014. What are developers talking about? an analysis of topics and trends in stack overflow. Empir. Softw. Eng. 19 (3), 619–654.

Bazelli, B., Hindle, A., Stroulia, E., 2013. On the personality traits of stackoverflow users. In: 2013 IEEE International Conference on Software Maintenance. IEEE, pp. 460–463.

Beyer, S., Macho, C., Penta, M.Di., Pinzger, M., 2020. What kind of questions do developers ask on stack overflow? A comparison of automated approaches to classify posts into question categories. Empir. Softw. Eng. 25 (3), 2258–2301.

Beyer, S., Pinzger, M., 2014. A manual categorization of android app development issues on stack overflow. In: 2014 IEEE International Conference on Software Maintenance and Evolution. IEEE, pp. 531–535.

Blei, D.M., Ng, A.Y., Jordan, M.I., 2003. Latent dirichlet allocation. J. Mach. Learn. Res. 3 (Jan), 993–1022.

Bmabey, 0000. Bmabey/pyldavis. Retrieved May 07, 2021, from https://github.com/bmabey/pyLDAVis.

Bosu, A., Corley, C.S., Heaton, D., Chatterji, D., Carver, J.C., Kraft, N.A., 2013. Building reputation in stackoverflow: an empirical investigation. In: 2013 10th Working Conference on Mining Software Repositories (MSR). IEEE, pp. 89–92.

Brem, A., Viardot, E., Nylund, P.A., 2021. Implications of the coronavirus (COVID-19) outbreak for innovation: Which technologies will improve our lives? Technol. Forecast. Soc. Change 163, 120451.

Carver, J.C., Kendall, R.P., Squires, S.E., Post, D.E., 2007. Software development environments for scientific and engineering software: A series of case studies. In: 29th International Conference on Software Engineering (ICSE'07). IEEE, pp. 550–559.

Celikyilmaz, A., Hakkani-Tur, D., Tur, G., 2010. LDA based similarity modeling for question answering. In: Proceedings of the NAACL HLT 2010 Workshop on Semantic Search pp. 1–9.

Chakraborty, P., Shahriyar, R., Iqbal, A., Uddin, G., 2021. How do developers discuss and support new programming languages in technical Q & A site? An empirical study of go, swift, and rust in stack overflow. Inf. Softw. Technol. 106603.

Chen, H., Coogle, J., Damevski, K., 2019. Modeling stack overflow tags and topics as a hierarchy of concepts. J. Syst. Softw. 156, 283–299.

Chen, C., Xing, Z., 2016. Mining technology landscape from stack overflow. In: Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement pp. 1–10.

Correa, D., Sureka, A., 2013. Fit or unfit: analysis and prediction of closed questions on stack overflow. In: Proceedings of the first ACM conference on Online social networks pp. 201–212.

Cox, M.A., Cox, T.F., 2008. Multidimensional scaling. In: Handbook of Data Visualization. Springer, Berlin, Heidelberg, pp. 315–347.

Cui, B., Yao, J., Cong, G., Huang, Y., 2010. Evolutionary taxonomy construction from dynamic tag space. In: International Conference on Web Information Systems Engineering (. Springer, Berlin, Heidelberg, pp. 105–119.

Diyanati, A., Sheykhahmadloo, B.S., Fakhrahmad, S.M., Sadredini, M.H., Diyanati, M.H., 2020. A proposed approach to determining expertise level of StackOverflow programmers based on mining of user comments. J. Comput. Lang. 61, 101000.

George, G., Lakhani, K.R., Puranam, P., 2020. What has changed? The impact of covid pandemic on the technology and innovation management research agenda. J. Manage. Stud. 57 (8), 1754–1758.

Georgiou, K., Mittas, N., Angelis, L., Chatzigeorgiou, A., 2020. A preliminary study of knowledge-sharing related to covid-19 pandemic in stack overflow. In: 2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA). IEEE Computer Society, pp. 517–520.

Georgiou, K., Papoutsoglou, M., Vakali, A., Angelis, L., 2019. Software technologies skills: A graph-based study to capture their associations and dynamics. In: Proceedings of the 9th Balkan Conference on Informatics. pp. 1–7.

Gruetze, T., Krestel, R., Naumann, F., 2016. Topic shifts in stackoverflow: Ask it like socrates. In: International Conference on Applications of Natural Language To Information Systems. Springer, Cham, pp. 213–221.

Johri, V., Bansal, S., 2018. Identifying trends in technologies and programming languages using topic modeling. In: 2018 IEEE 12th International Conference on Semantic Computing (ICSC). IEEE, pp. 391–396.

Kaplan, E.L., Meier, P., 1958. Nonparametric estimation from incomplete observations. J. Amer. Statist. Assoc. 53 (282), 457–481.

Kelion, L., 2020. Coronavirus: First google/apple-based CONTACT-tracing app launched. Retrieved April 10, 2021, from https://www.bbc.com/news/technology-52807635.

Kelly, D., 2015. Scientific software development viewed as knowledge acquisition: Towards understanding the development of risk-averse scientific software. J. Syst. Softw. 109, 50–61.

Kleinbaum, D.G., Klein, M., 2012. Survival Analysis: A Self-Learning Text, third ed. Springer.

Kumar, A., Gupta, P.K., Srivastava, A., 2020. A review of modern technologies for tackling COVID-19 pandemic. Diabetes Metab. Syndrome: Clin. Res. Rev. 14 (4), 569–573.

Linares-Vásquez, M., Dit, B., Poshyvanyk, D., 2013. An exploratory analysis of mobile development issues using stack overflow. In: 2013 10th Working Conference on Mining Software Repositories (MSR). IEEE, pp. 93–96.

Mamykina, L., Manoim, B., Mittal, M., Hripcsak, G., Hartmann, B., 2011. Design lessons from the fastest q & a site in the west. In: Proceedings of the SIGCHI conference on Human factors in computing systems. pp. 2857–2866.

Meldrum, S., Licorish, S.A., Owen, C.A., Savarimuthu, B.T.R., 2020. Understanding stack overflow code quality: A recommendation of caution. Sci. Comput. Programm. 199, 102516.

Movshovitz-Attias, D., Movshovitz-Attias, Y., Steenkiste, P., Faloutsos, C., 2013. Analysis of the reputation system and user contributions on a question answering website: Stackoverflow. In: 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2013). IEEE, pp. 886–893.

Neshati, M., 2017. On early detection of high voted q & a on stack overflow. Inf. Process. Manage. 53 (4), 780–798.

Nguyen-Hoan, L., Flint, S., Sankaranarayana, R., 2010. A survey of scientific software development. In: Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement. pp. 1–10.

Niraula, N., Banjade, R., Ştefănescu, D., Rus, V., 2013. Experiments with semantic similarity measures based on lda and lsa. In: International Conference on Statistical Language and Speech Processing. Springer, Berlin, Heidelberg, pp. 188–199.

Ortega, F., Convertino, G., Zancanaro, M., Piccardi, T., 2014. Assessing the performance of question-and-answer communities using survival analysis. arXiv preprint arXiv:1407.5903.

Papoutsoglou, M., Kapitsaki, G.M., Angelis, L., 2020. Modeling the effect of the badges gamification mechanism on personality traits of Stack overflow users. Simul. Model. Pract. Theory 105, 102157.

Ponzanelli, L., Mocci, A., Bacchelli, A., Lanza, M., Fullerton, D., 2014. Improving low quality stack overflow post detection. 2014 IEEE International Conference on Software Maintenance and Evolution 541–544.

Raban, D.R., 2009. Self-presentation and the value of information in Q & A websites. J. Am. Soc. Inf. Sci. Technol. 60 (12), 2465–2473.

Rosen, C., Shihab, E., 2016. What are mobile developers asking about? a large-scale study using stack overflow. Empir. Softw. Eng. 21 (3), 1192–1223.

Rus, V., Niraula, N., Banjade, R., 2013. Similarity measures based on latent dirichlet allocation. In: International Conference on Intelligent Text Processing and Computational Linguistics. Springer, Berlin, Heidelberg, pp. 459–470.

Segal, J., Morris, C., 2008. Developing scientific software. IEE Softw. 25 (4), 18–20.

Selenium, 0000. Retrieved April 10, 2021, from https://pypi.org/project/selenium/.

Shao, B., Yan, J., 2017. Recommending answerers for stack overflow with lda model. In: Proceedings of the 12th Chinese Conference on Computer Supported Cooperative Work and Social Computing 80–86.

Sievert, C., Shirley, K., 2014. LDAvis: A method for visualizing and interpreting topics. In: Proceedings of the workshop on interactive language learning, visualization, and interfaces. pp. 63–70.

Somasundaram, K., Murphy, G.C., 2012. Automatic categorization of bug reports using latent dirichlet allocation. In: Proceedings of the 5th India Software Engineering Conference, pp. 125–130.

Stack exchange, 0000. Retrieved April 10, 2021, from https://stackexchange.com/sites#technology.

Stack overflow developer SURVEY 2020, 0000. Retrieved April 10, 2021, from https://insights.stackoverflow.com/survey/2020.

Tong, Z., Zhang, H., 2016. A text mining research based on LDA topic modelling. In: International Conference on Computer Science, Engineering and Information Technology pp. 201–210.

Treude, C., Barzilay, O., Storey, M.A., 2011. How do programmers ask and answer questions on the web? (NIER track). In: Proceedings of the 33rd International Conference on Software Engineering, pp. 804–807.

Vaishya, R., Javaid, M., Khan, I.H., Haleem, A., 2020. Artificial intelligence (AI) applications for COVID-19 pandemic. Diabetes Metab. Syndrome: Clin. Res. Rev. 14 (4), 337–339.

Venkatesh, P.K., Wang, S., Zhang, F., Zou, Y., Hassan, A.E., 2016. What do client developers concern when using web apis? an empirical study on developer forums and stack overflow. In: 2016 IEEE International Conference on Web Services (ICWS). IEEE, pp. 131–138.

Villanes, I.K., Ascate, S.M., Gomes, J., Dias-Neto, A.C., 2017. What are software engineers asking about android testing on stack overflow? In: Proceedings of the 31st Brazilian Symposium on Software Engineering pp. 104–113.

Wang, S., Chen, T.H.P., Hassan, A.E., 2018a. How do users revise answers on technical q & a websites? A case study on stack overflow. IEEE Trans. Softw. Eng..

Wang, S., Chen, T.H., Hassan, A.E., 2018b. Understanding the factors for fast answers in technical Q & A websites. Empir. Softw. Eng. 23 (3), 1552–1593.

Wang, S., Lo, D., Jiang, L., 2013. An empirical study on developer interactions in stack overflow. In: Proceedings of the 28th Annual ACM Symposium on Applied Computing 1019–1024.

Wang, L., Zhang, L., Jiang, J., 2020. Duplicate question detection with deep learning in stack overflow. IEEE Access 8, 25964–25975.

Westwood, S., Johnson, M., Bunge, B., 0000. Predicting programming community popularity on stackoverflow from initial affiliation networks.

Wilson, G., 2006. Software carpentry: getting scientists to write better code by making them more productive. Comput. Sci. Eng. 8 (6), 66–69.

Yang, X.L., Lo, D., Xia, X., Wan, Z.Y., Sun, J.L., 2016. What security questions do developers ask? a large-scale study of stack overflow posts. J. Comput. Sci. Tech. 31 (5), 910–924.

Ye, D., Xing, Z., Kapre, N., 2017. The structure and dynamics of knowledge network in domain-specific q & a sites: a case study of stack overflow. Empir. Softw. Eng. 22 (1), 375–406.

Zou, J., Xu, L., Yang, M., Zhang, X., Yang, D., 2017. Towards comprehending the non-functional requirements through developers' eyes: An exploration of stack overflow using topic analysis. Inf. Softw. Technol. 84, 19–32.