Contents lists available at ScienceDirect

Future Generation Computer Systems

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

Understanding the social evolution of the Java community in Stack Overflow: A 10-year study of developer interactions

Guillermo Blanco [a,b,c], Roi Pérez-López [a,b,c], Florentino Fdez-Riverola [a,b,c], Anália Maria Garcia Lourenço [a,b,c,d,*]

[a] *ESEI - Department of Computer Science, University of Vigo, Edificio Politécnico, Campus Universitario As Lagoas S/N 32004, Ourense, Spain*
[b] *CINBIO - Centro de Investigaciones Biomédicas, University of Vigo, Campus Universitario Lagoas-Marcosende, 36310, Vigo, Spain*
[c] *SING Research Group, Galicia Sur Health Research Institute (IIS Galicia Sur), SERGAS-UVIGO, Vigo, Spain*
[d] *CEB - Centre of Biological Engineering, University of Minho, Campus de Gualtar, 4710-057 Braga, Portugal*

ARTICLE INFO

Article history:
Received 30 April 2019
Received in revised form 28 October 2019
Accepted 13 December 2019
Available online 17 December 2019

Keywords:
Q&A platforms
Stack Overflow
Java
Quality of service
Community analysis
Machine learning

ABSTRACT

Today, Social Media is a key information source for a wide range of domains as a means to gain a better understanding of information flows and user communities. This work introduces a methodology combining machine learning and graph mining approaches to address relevant aspects of quality of service and user intrinsic motivation from the user's perspective. The focus of the present analysis is set on the social interactions among software developers via Stack Overflow. Over the last 10 years, software developers have become intensively involved in knowledge sharing and platforms such as Stack Overflow have accumulated a lot of development data and knowledge. The proposed methodology is applied to explore the social dynamics of the Java programming language community and bring forward relevant, non-trivial knowledge about developer interests, information flows and user engagement and reputation. The ultimate aim is to improve question preparation towards better question routing and voting outcome.

1. Introduction

In the last decades, community-based question-and-answer (Q&A) sites have become very popular and have enabled knowledge sharing at unprecedented levels. Stack Overflow is the *de facto* Q&A website for topics in Computer Science. Current platform statistics account for 10 million users, 18 million questions, and 27 million answers (71% of questions answered) [1]. Moreover, the number of programming languages in use has increased. In 2018, the Developer's Survey of Stack Overflow listed 38 different programming languages within the most loved, dreaded, and wanted languages.

Recently, Stack Overflow released its posts as online archives (https://archive.org/download/stackexchange), which paved the way to streamline the analysis of these conversation threads in various useful ways.

The present work complements current literature by introducing a new methodology of analysis that tackles the quality of service and user reputation in Q&A platforms from the user's perspective. More specifically, this methodology aims to improve the user experience by proposing practical recommendations on how the user can point his questions in the right direction, i.e. reducing the number of questions with no answers, routing questions to the right answers, and promoting the content quality of the platform by identifying low-quality contents. As a meaningful case study, this paper explores the social evolution experienced by the Java developer community on Stack Overflow, i.e. an in-depth look into the topics that have motivated more discussion over the years, the evolving of social dynamics, including user altruism and reputation, and the cross-reference of internal contents as well as external sources.

To the best of our knowledge, such an integrative analysis has not been presented before. A number of works exist addressing similar topics though. The related work section describes some of these works while pinpointing the new, hereby presented contributions.

2. Related work

Understanding the dynamics of participation in Q&A platforms is essential to improve the value of crowdsourced knowledge and

[*] Corresponding author at: ESEI - Department of Computer Science, University of Vigo, Edificio Politécnico, Campus Universitario As Lagoas S/N 32004, Ourense, Spain.

E-mail addresses: guillermo@guillermoblanco.es (G. Blanco), rplopez@esei.uvigo.es (R. Pérez-López), riverola@uvigo.es (F. Fdez-Riverola), analia@uvigo.es (A.M.G. Lourenço).

URLs: http://sing-group.org/ (G. Blanco), http://sing-group.org/ (R. Pérez-López), http://sing-group.org/riverola (F. Fdez-Riverola), http://sing-group.org/ (A.M.G. Lourenço).

https://doi.org/10.1016/j.future.2019.12.021
0167-739X/© 2019 Elsevier B.V. All rights reserved.

the quality of service as well as to promote user engagement. Many works have focused on the platform's needs and challenges, but few works address the user's perspective, namely the duality of being information seekers and information producers.

The increasing number of low-value, unanswered questions has prompted the need to learn how to pose well-received questions [2]. Question routing, namely identification of best answers and the identification of similar questions, is also essential to ensure the quality of contents, avoid post duplication and optimize the use of internal resources (i.e. redirecting to previously vetted answers) [3]. Calefato et al. proposed a framework of factors influencing the success of questions in Stack Overflow, notably factors that can be acted upon by software developers when writing a question to ask for technical help [4].

The present work proposes a semantic matching topic model–model approach to question routing. The analysis of question-and-answer questions and, more specifically, the application of natural language processing as an integral part of machine learning has been explored in various ways [5]. For example, the mining of questions about design patterns has been pursued to develop a knowledge base about problem-prone patterns [6], the recognition of mentions to architecture-relevant and technology-related information was pursued with the purpose of knowledge structuring [7], and the construction of probabilistic models to capture the correlation between natural language textual descriptions and code snippets enabled the implementation of query systems supporting code retrieval and synthesis [8]. The mining of code snippets in combination with textual information is being applied to the categorization of posts by programming language [9] and the downside of code sharing, caused by bug propagations, software maintenance issues, and software licence violation issues, is also being explored [10]. At another level, the implementation of recommendation systems fed by Stack Overflow threads has been proposed as a means to enable IDE programming prompters that can help developers to perform searches (and evaluate results) without interrupting their workflow [11–13]. Similarly, summarization techniques and pattern-based techniques have been applied to automatically augment API documentation, namely in terms of concepts, purpose, usage scenarios, and code examples [14].

Regardless the specifics of the work, the majority of previous approaches have focused on evaluating content quality after the fact, i.e. after questions have been resolved, whereas the goal of the present work is to be able to evaluate the quality of the question and the probability of being successfully closed prior to the submission. Likewise, many existing methods try to tackle question routing by learning user model from structure and topic information, whereas the present method proposes the analysis from the viewpoint of knowledge graph embedding. So, the present contribution lays on how questions and answers are categorized and their interrelation is semantically explored. In particular, the two-fold categorization of questions, based on title and body contents, and the categorization of answers are proposed as means to evaluate the quality of questions and answers individually as well as explore their semantic interrelation. From user's perspective is important to be aware of how the quality of the title affects the routing of the question and to what extent the different contents of the body can impact the discovery of similar questions. Equally important, the hereby proposed methodology manages user expectations, raising awareness of the quality of topic-specific contents and the expected, average time of question-solving.

Another important part of knowledge sharing is the reference of URLs of web resources in question and answer posts. External Web resources, i.e. URLs external to stackoverflow.com, include API documents and language references, as opposed to internal web resources, i.e. URLs within stackoverflow.com, which are cross-referenced to other posts. Some work has been devoted to the construction of a knowledge base of web resources that have been shared on Stack Overflow [15]. This included the use of predictive models to distinguish URLs of official documentation from URLs of other types of web resources. While sharing some goals of analysis, the present work focuses on the characterization of novelty, difficulty and interest of the questions, that is how likely it is that the questions of a given topic can be solved without resorting to external resources. That is, topics that are recurrently addressed via external documentation are likely to have few and/or less-experienced contributors.

Research on the promotion of user intrinsic motivation, as proposed in this work, complements work on the enhancement of user experience in a significant way. Most users in Q&A platforms come from search engine result pages, and their motivation in contributing knowledge through the platform are self-presentation, recognition, and social learning opportunities. Stack Overflow, like other platforms, devised a reputation and badge-earning system to engage users as active contributors rather than just posting questions and seeking answers. The reputation, like the score of members, reflects the involvement of the user and the quality of user's questions and answers. And the badges acknowledge the user's efforts to successfully answer questions. So far, research has focused on gaining a deeper, richer understanding of user motivations. For example, work has been done to correlate the delay of and reputation for a given answer [16] and to understand why almost half of the Stack Overflow users asks only one question during their membership [17]. Here, motivation is modelled at the topic level, taking into consideration the various interests of developers and how keen a developer may be to contribute to some topics rather than others and, most notably, altruism is seen both like the quality of the social group as well as a personality trait.

3. Materials and methods

3.1. Data retrieval and preparation

Conversation threads tagged as Java-related and posted from 2008 till 2018 were downloaded through the Stack Overflow archives (https://archive.org/details/stackexchange). This amounted in a total of 3.33 million posts, from which 1.8 million represented questions and, within this set, approximately 0.9 million had answers (i.e. closed questions, referred here as Q&A pairs). These communication threads were sustained by a total of 0.3 million unique users.

The textual information in Q&A pairs, i.e. the title and body of the question as well as the body of the answer, were cleaned and prepared for analysis. Some pre-processing steps were required to clean the text. Tokenization and the filtering of stopwords (including English stopwords and generic programming-related stopwords) and too short words (i.e. 2 or fewer chars) were performed using the Natural Language Tool Kit (NLTK) package (https://www.nltk.org/) [18]. The libraries BeautifulSoup, (https://www.crummy.com/software/BeautifulSoup) and Twitter-text-python (https://github.com/edmondburnett/twitter-text-python) were applied in HTML parsing. POS tagging, namely the selection of nouns and verbs as most content-bearing contents, was enabled by the spaCy library (https://spacy.io/) [19]. The same library was also used to lemmatize these tokens. Moreover, the generation of uni-, bi- and tri-grams (hereby generally called terms) and the calculation of the corresponding TF-IDF frequencies was performed with the Gensim library [20]. The hyperparametrization of the values of min count and threshold enabled the selection of the most content-bearing terms. In particular, the hyperparameter min count defined the number of times that a unigram needed to appear in the set of documents to be included in the corpus, while the threshold established the minimum value of term frequency-inverse document frequency (TF-IDF) required to take a bigram or trigram into account.

Stack Overflow metadata, namely user-level information (e.g. user id) and question–answer-specific data (e.g. answer score) were also retrieved as complements of the textual contents analysis.

3.2. Topic discovery

The Gensim implementation of the Latent Dirichlet Allocation (LDA) method was applied to uncover latent topics in the questions [21]. Specifically, question titles and bodies were represented using a three-level hierarchical Bayesian model.

The hyperparametrization of the Dirichlet distribution attributes was conducted to obtain an optimal set of parameters for the topic modelling of the question titles as well as question bodies. Specifically, hyperparametrization was issued for the number of topics, the question-topic density α (i.e. a higher α implied that the question should contain more topics) and the topic-word density β (i.e. a higher β implied that topics should contain most of the terms in the questions).

The relevance of term w to topic k was given by:

$$relevance\,(w,k|\lambda) = \lambda \times \log(\phi_{kw}) + (1-\lambda)\log(\frac{\phi_{kw}}{p(w)})$$

where $0 \leq \lambda \leq 1$ determined the weight given to the probability of term w under topic k relative to its lift (i.e. how prevalent the term was across all topics) [22]. That is, $\lambda = 1$ ranked terms in decreasing order of their topic-specific probability, and $\lambda = 0$ ranked terms based only on their lift.

Term distinctiveness and term saliency described how informative the term w was for determining the generating topic, versus a randomly selected term in an information-theoretic sense [23]. Distinctiveness was evaluated based on the Kullback–Leibler divergence [24] between the conditional probability $P(T|w)$, i.e. the likelihood that the observed term w was generated by the latent topic T, and the marginal probability $P(T)$, i.e. the likelihood that any randomly selected term w was generated by topic T:

$$distinctiveness\,(w) = \sum_{T} P\,(T|w)\,log\,\frac{P(T|w)}{P(T)}$$

And, term saliency was defined by the product:

$$saliency\,(w) = P(w) \times distinctiveness(w)$$

If the term w occurred in all topics, w was not informative about the topical mixture of the document and thus, it would have a low distinctiveness score. So, saliency enabled faster differentiation among the topics and the identification of potential "junk topics", i.e. topics lacking salient terms.

The interactive, web-based visualization of the LDA models, which supported manual inspection and the discussion of the results, was possible using the pyLDAvis library (https://pypi.org/project/pyLDAvis/2.1.1/) [22].

3.3. Graph mining

Graphs offer an intuitive and visually appealing means to represent and analyse relations among objects or entities. Although social network analysis is typically associated with user interactions, graphs enabled a broader spectrum of representations and mining.

Two graph models support of the present analysis, namely a bipartite graph represented the relations between question topics and URLs, and a directed graph depicted user communications. Both graphs were generally described in terms of the corresponding number of nodes and edges, as well as metrics of degree, characteristic path length, clustering coefficient and the average number of neighbours [25]. Graph connectedness was described in three ways [26–28]: degree centrality measured the total amount of direct links with the other nodes (i.e. higher degree implies the node is more central), betweenness centrality depicted the role of nodes as mediators (i.e. if others nodes have to go through the node to ensure communication, then the node has a high betweenness centrality), and closeness centrality measured the convenience and ease of connections between each node and the rest of nodes (i.e. if the average shortest path of the node is small, then the node has a high closeness centrality).

Finally, the clustering coefficient measured the degree to which nodes tended to cluster together, i.e. the likelihood of link density be greater than the average probability of randomly establishing a link between two nodes [29].

The graphs were analysed using Gephi version 0.9.2 [30] and Cytoscape version 3.7.1 [31].

4. Results and discussion

The study of the social interplay of the Java community in Stack Overflow throughout the last decade enabled the evaluation of the proposed methodology, in terms of correctness and robustness, as well as scalability in practical domains. The next sections describe the community evolution in general terms and then, the modelling of Q&A contents and the modelling of user intrinsic motivation.

Contents modelling aims to bring forward valuable, actionable information towards improving question preparation and routing, reducing the number of questions that remain unsatisfied or are satisfied very late, and thus, enhance user experience. Conversely, the modelling of user motivation addresses important features about the relations, autonomy, mastery and purpose of user communications.

4.1. Social dynamics evolution

The Java programming language has evolved considerably in the last decade, both in terms of user engagement and technological advancements. Fig. 1 portraits such evolution in terms of the volume of questions posted on Stack Overflow and the volume, score and response time of the contributed answers.

The increment of the number of questions and answers over time is noticeable. Yet, looking more closely, the proportion of answers per the number of issued questions has decreased in the most recent years. This can be justified by the knowledgeability of Java developers, i.e. users are ever more familiar with Java and fewer answers are needed to find one that closes the question. Equally, and except for 2017, the average time to close a question was fairly constant and long (between 10 and 20 days) throughout the period under analysis, which denotes that this community lacks socializers and philanthropists willing to share knowledge and boost information flows. Also noteworthy, the average score per answer decreased over the years. This can be motivated by two main reasons. First, the users are becoming less "generous" (e.g. poor quality answers or answers basically pointing to other resources). Second, questions are being closed more quickly and thus, answers are trending less time in Stack Overflow pages.

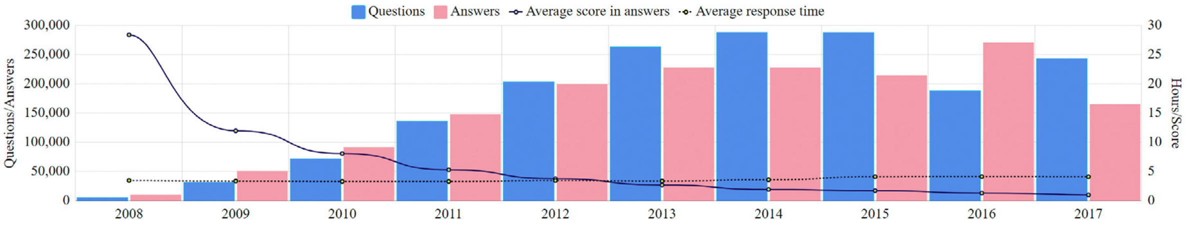

Fig. 1. Evolution of the Java community throughout the years.

4.2. Contents modelling: Q&A topic modelling

Question titles are supposed to be informative about the nature/goal of the questions and thus, they should be sufficient to determine the topic of the questions. However, this assumption fails whenever titles are too short or generic and, under such circumstances, the processing of question bodies is required to be able to rout the question. In the case of Java posts, after text processing, most of the question titles were reduced to 2 to 4 content-bearing terms whereas question bodies entailed between 6 and 30 terms (Fig. 2).

The analysis of question contents entailed three steps, i.e. the modelling of topics in question titles, the modelling of topics in question bodies, and the alignment of both models to obtain an integrative perspective of the needs/doubts of the user towards finding the right subcommunity of developers (i.e. users interested in similar topics) and thus, increasing the probability of obtaining a suitable answer (including the matching of question with similar, previously answered questions as well as the motivation to answer question about the topic).

The optimal Dirichlet distribution attributes for the topic modelling of the question titles were determined based on the coherence and perplexity per topic, i.e. a total of 10 topics, considering α as "symmetric" (i.e. the prior is parameterized by the alpha vector such that there is one parameter per expected topic) and $\beta =$ "1.0" (Fig. 3).

Topic labelling was possible via manual inspection of the salient terms and the exploration of some of the members (i.e. questions). For example, the most salient terms of the topic "Types of files and data" were "read", "xml", "write" and "string", whereas "button", "field", "hibernate", "query" and "view" represented the topic "Display views and database", and "problem", "exception", "error", "throw" and "miss" represented the topic "Issues". Detailed information about this topic model can be found in Supplementary Material 1.

Following a similar process of analysis, the optimal Dirichlet distribution attributes for the topic modelling of the question bodies were deemed to be 25 topics, with symmetric α and no β (i.e. a term could belong to more than one topic). For the sake simplicity, and taking into account language specifics, the topics were manually divided into 8 families. Fig. 4 depicts the topical distribution of these topical families. Noteworthy, adjacent clusters usually correspond to topics belonging to the same family. For example, the topical family "Security" embrace the topics "Authentication", which was based on the terms "register", "login" "username password" and "session", and the topic "Encrypt", which was based on the terms "encrypt", "decrypt" and "codec". Supplementary Material 1 offers a detailed description of all the topics and topical families.

Next, the topic models of the titles and the bodies of the questions were enriched with the quantitative characterization of the corresponding answers. Specifically, such per topic characterization entailed features such as popularity (based on the number of Q&A pairs), the average time to answer a question, the percentage of answers that included code snippets and the percentage of answers that contained URLs.

Supplementary Material 1 presents the full characterization of the topics. The information is displayed as a matrix, that is columns represent title topics and rows represent body topics and cells depict the metric under observation. The hue colour represents the magnitude of the value, i.e. the most intense green identifies the best value and the most intense red the worst value. Therefore, the worksheets "posts per topic", "answers per topic", "scores per topic", "answer times per topic", "code" and "URL" describe topic pair occurrences such that "answers per topic" depicts the answer count corresponding to the topic pair, "scores per topic" contains the average score of accepted answers for the topic pair, and "answer times per topic" contains the average time that takes to answer a question belong to the topic pair. Moreover, the last two worksheets describe the percentage of code snippets in the answers and percentage of answers that contained URLs.

4.3. Contents modelling: question categorization and predicted answer statistics

The topic pairs of question titles and bodies were categorized based on the previously calculated metrics, i.e. popularity, average answer time, code probability and URL probability. The aim was to be able to predict if a new incoming question belongs to a low profile topic or a mainstream topic, and therefore, whether it is likely to be answered quickly or more slowly. Likewise, this categorization was considered of help to predict how likely it is that the answer will contain code snippets or URLs.

Normal distribution was assumed for the data of all metrics and was used to further categorize the Q&A pairs falling into each topic. As described in Table 1, metric values were generally classified as low (i.e. value in the 25th percentile), medium (i.e. between the 25th percentile and the 75th percentile) and high (i.e. value in the 75th percentile).

Overall, these results were consistent with the average number of answers including code snippets (which was between 0.65 and 0.77), which indicated that most users typically post code snippets rather than textual explanations. Also interesting, URL mentions were often included as further documentation of the provided code snippets.

The bipartite graph representing the connectivity of the answers to internal or external URLs is shown in Fig. 5. For simplicity's sake, only the top 15 most occurring URLs per topic ae depicted (i.e. a total of 28 different URLs). The opacity of edge colour illustrates the contribution of the topic to the node degree of URL, while the node degree of the URL represents overall contributions. The URLs with the greatest in-degree belonged to "stackoverflow.com" (i.e. to 8.08% and in-degree of 10) and "github.com" (22.8% and in-degree of 10). Other Q&A platforms and documentation resources were also within the top 15 per topic, such as "docs.oracle.com" (15.85% and in-degree of 10) and developer.android.com (10.7% and indegree of 10). Some of the URLs of lower in-degree are also of interest, since they were specific of one topic. For example, "issues.apache.org" and "msdn.microsoft.com" belonged to "Issues" "Client-Server" topics, respectively.

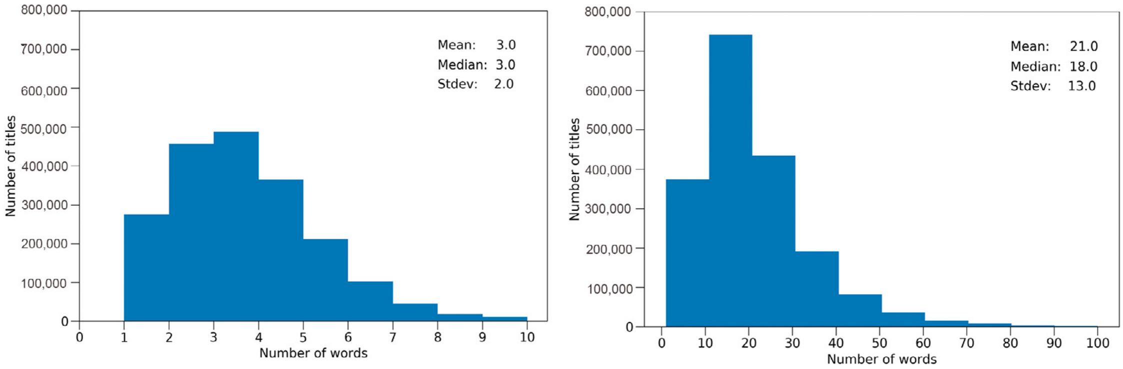

Fig. 2. Histograms of content-bearing terms in the titles and bodies of Java questions.

Table 1
Normal distribution of answer metrics.

Metric	Low (25th percentile)	Medium (between 25th and 75th percentiles)	High (75th percentile)
Answering time	Slow $\cong$ [hours $<$ 76]	Normal $\cong$ [76 $\geq$ hours $\leq$ 308]	Quick $\cong$ [hours $>$ 308]
Popularity	Low $\cong$ [questions $<$ 78]	Normal $\cong$ [78 $\geq$ questions $\leq$ 5715]	Mainstream $\cong$ [questions $>$ 5715]
Code percentage	Low $\cong$ [code $<$ 65%]	Normal $\cong$ [65% $\geq$ code $\leq$ 77%]	High $\cong$ [code $>$ 77%]
URL percentage	Low $\cong$ [URL $<$ 17%]	Normal $\cong$ [17% $\geq$ URL $\leq$ 24%]	High $\cong$ [URL $>$ 24%]

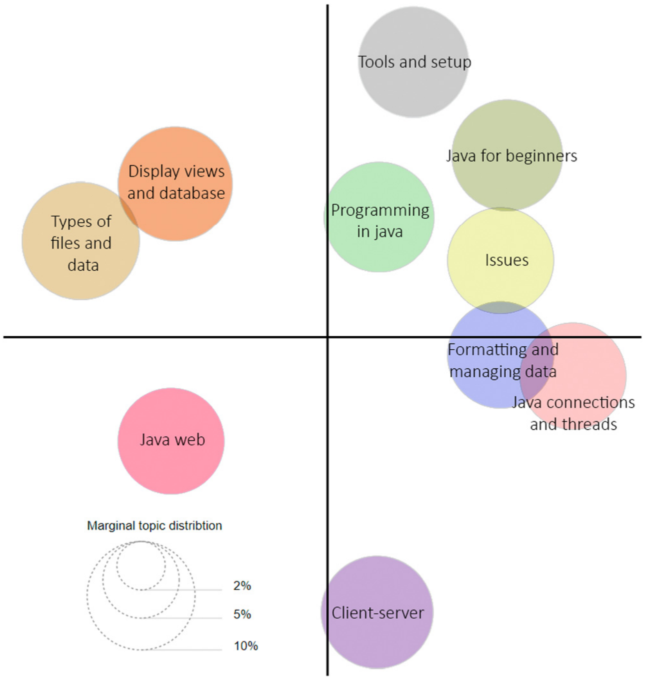

Fig. 3. Intertopic distance map of the model constructed for the question titles.

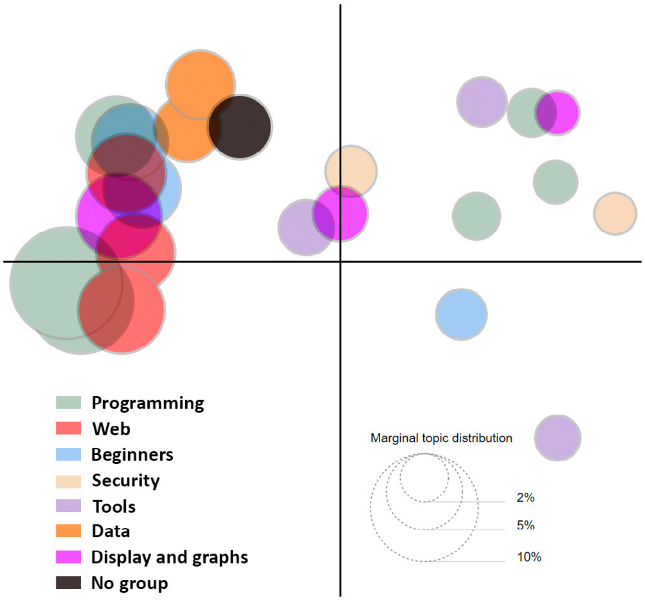

Fig. 4. Intertopic distance map of the topic model constructed for the question bodies.

Furthermore, the analysis of URLs in answers was expanded with the knowledge of the URLs most used per question topic. The "github.com" was the most referred URL in all the topics, but it was interesting to study the diversity of the rest of the referred URLs. For example, the answers of the "Display views and database" topic contained 7.4% of the URLs mentioned in the entire set of answers, and the second-most linked URL was "developer.android.com". In turn, the answers of the "Tools and setup" topic included 11.9% of URLs, and the second most linked URL was "springframework.com".

4.4. Modelling user intrinsic motivation: user reputation and community demography

In Stack Overflow, users earn their reputation through active participation in the community, e.g. by posing well-received questions and providing helpful answers. After achieving a high reputation level, a user is allowed to up-vote or comment on questions and answers, and even edit posts of other users. This reputation is represented in the form of badges, i.e. "bronze" (i.e above 100 score in at least 20 answers), "silver" (i.e. above 400 score in at least 80 answers), and "gold" (i.e. above 1000 score in at least 200 answers). Users can be awarded badges for general platform participation or can be awarded for contribution in specific language programming communities. Here, a total of 333,344 unique users were identified in the Java community, from which 5460 had bronze badges, 1405 had silver badges and 502 had gold badges.

Graph mining was applied to represent social interplay and be able to navigate within and across subcommunities with particular personality traits. In particular, Fig. 6 shows that the graph representing the most active users in the Java community has a subgraph of densely connected users while the rest of users exhibited just a few interactions. The degree of the nodes corresponds to answer activity on Stack Overflow. Due to the size of the graph, and to improve interpretation ability, edge colour indicates the user that sent the answer, whereas the colour of the nodes depicts the user group, which is determined by the proportional difference of answer activity on Stack Overflow (i.e. greener nodes indicate users with low activity while redder nodes indicate users with high activity).

As one may observe in the displayed graph, there is a huge difference between users in terms of activity, i.e. a few users (0.4% of the set) answer most of the questions (red coloured nodes). For example, the user 22 656 (aka Jon Skeet which node obtained a betweenness centrality of 0.000529 and indegree of 6) has 1,139,226 score of reputation and the user 157882 (aka BalusC which node obtained a betweenness centrality of 0.014 and indegree of 6) has a score of reputation of 893,495 according to Stack Overflow metrics. In turn, the users in green coloured nodes represented more than 90% of the total of users. Therefore, this system of reputation evaluation was deemed insufficient to classify users.

The proposed methodology complements the platform's user classification system by introducing more features of intrinsic motivation into the calculation and analysing user participation topic-wise. In particular, the proposed classification system accounts for user implication in each topic (i.e. the frequency of user's questions and answers falling in the topic). Calculations per topic included the user cumulative answer score (i.e. classification of users as beginners or experts), the level of altruism (i.e. the proportion of answers of the user over his questions when the proportion is over 1) and the level of selfishness (i.e. the proportion of questions of the user over his answers when the proportion is above).

Users were categorized according to their topic-specific motivation (Normal distribution). Table 2 describes the threshold values for badge, altruism and expertise categorization. The level of altruism was based on question and answer proportion, i.e. if the number of questions was higher than the number of answers, the user was categorized as selfish and was considered altruist otherwise. The level of expertise was based on the answer score average, being categorized as beginner a user with a low average score, and as expert when the average score was high.

The subset of users densely connected (coloured green in Fig. 6) was categorized according to these metrics. For example, the user 1844392 (aka piyush) was classified by Stack Overflow as a bronze user, and with the proposed metrics it is possible to detail his profile as being a beginner user (1.44 average score) with an altruist trait (79 altruism score). Likewise, the user 21234 (aka skaffman) is classified as gold user by the platform, and the proposed metrics picture him as an expert contributor (11,07 score) with altruist trait (32 altruism score).

4.5. Proof-of-concept: simulating methodology outcomes over new questions

The proof-of-concept of the practical application of the proposed methodology was conducted over a set of previously unseen, new incoming Java questions (some examples in Table 3). The experiments were designed to determine how accurate the proposed methodology is to categorize the question and to predict the probability of obtaining a valid, good quality answer within a reasonable time frame.

For example, the question with ID 48047128, which is entitled "Error 404 on SpringMVC using Maven", was categorized by the title-based topic model as belonging to the topic "Tools and setup" while the body-based model identified it as belonging to "Web basics" topics, i.e. into the "Web" family (the body started with "I keep getting this Error 404 and I've done/looked at almost all the solutions on StackOverflow that are relevant and still haven't fixed my problem. Why am I having this error?...."). By aligning these two categorizations, it is possible to provide predictions about the expected question visibility and answer (e.g. this question was deemed of normal popularity and of late time response) as well as the user demography of the subcommunity most likely to address the question (i.e fairly altruistic and gold badged users).

Another example is question ID 48043536, which is entitled "How to Load Image from a URL which generates random captcha image. Here the URL does not refer to a specific image but a dynamically generated image" was deemed mainstream for the pair "Type files and data" and "Web" and 60% of the gold users are likely to contribute to the topic pair. From the three examples in the table, the proposed methods predicts that the users that may potentially comment on the third question are the less altruistic. Notably, more than half of the users contributing to the topic are not altruist nor selfish.

These examples expose the added value of the categorization of question titles and bodies as well as of the proposed question and user metrics. More specifically, the popularity of the topic, based on how many questions currently have been classified in it, the corresponding average number of comments, the average time for a question to have a correct answer, and the probability of getting an URL or code snippet in that answer giving a detailed insight into the subcommunity more likely to address the question. Noteworthy, Stack Overflow indicates that the third question is likely to have a large participation of gold users , but the analysis of response scores shows a lower expertise level.

Supplementary Material 3 describes in detail the data used in the proof-of-concept.

5. Conclusions and future work

This paper presents a methodology that combines machine learning and graph mining techniques to analyse the quality of service and user reputation of communities in Q&A platforms. To be able to grasp how to formulate questions properly is beneficial not only for the information seekers, because it increases the likelihood of receiving support, but also for the whole community, since it enhances effective knowledge-sharing behaviour, and, most notably, the creation of long-lasting value pieces of knowledge.

The practical relevance of the proposed methodology is demonstrated in a 10-year study of developer interactions within the Stack Overflow Java community. The dataset under analysis included 1.8 M questions, with half a million Q&A pairs, and 333,344 unique users. An initial temporal analysis depicted how Java posts have evolved throughout the years, namely in terms of number of questions, number of answers, average answer score, and average response time.

One main focus of the study was the contents of Q&A pairs, i.e. understanding how Java developers of every level pose questions about the language and which topics have been attracting more attention throughout the years. Natural language processing was applied to the textual contents of the questions and answers to select the most content-bearing contents for further mining,

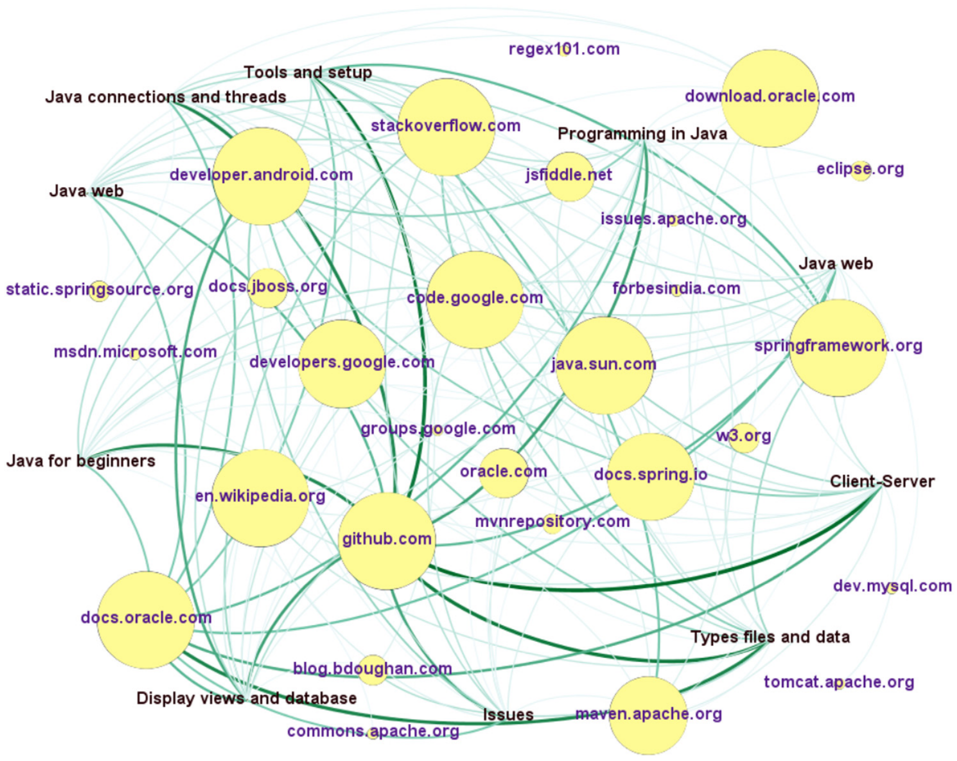

Fig. 5. Graph representing the relationships among topics and URLs (yellow nodes), opacity of edges corresponds to topic contribution to URL degree.

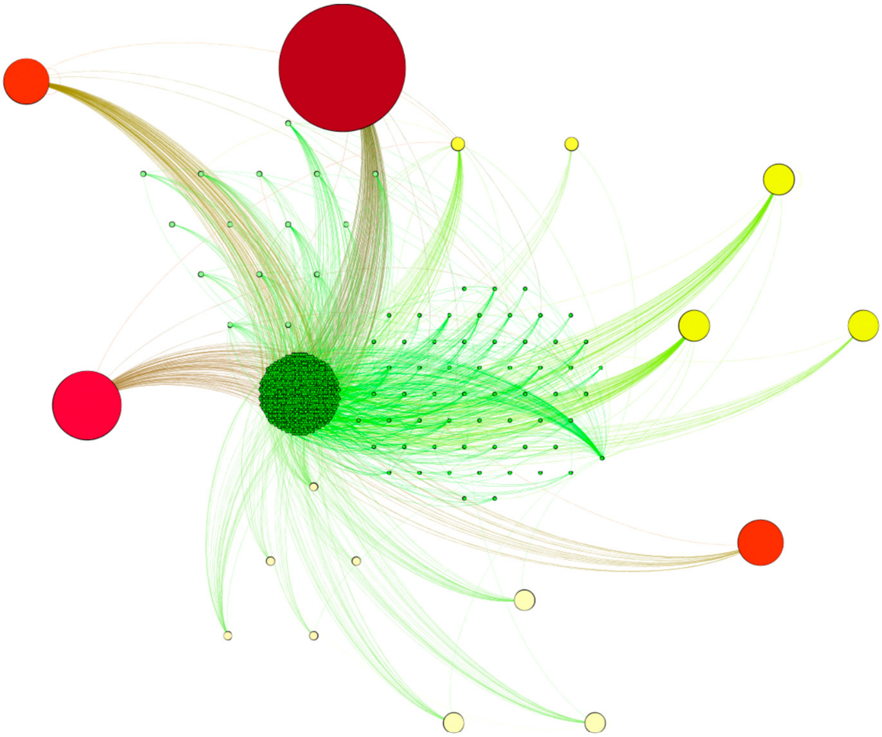

Fig. 6. Graph representing the most active users in the Java community. The size and colour of the nodes indicate the number of provided answers and the colour of the edges indicates the category of the user who sent the answer.

while LDA modelling was applied to group questions titles and bodies in meaningful topics. Types of files and data, Tools and setup, Java web or Issues were identified as title topics, while on the body classification it was necessary to extend the division to grouping the topics in family, "Web" (which belong to "Services" or "Web database"), "Beginners" (which belong to "Need code" or "Object oriented programming") or "Security" (which belong to "Authentication" and "Encrypt") were some of the identified families. Graph mining supported further analysis of social contents.

Table 2
Categorization of users.

Metric	Low	Medium	High
Stack overflow level	Bronze	Silver	Gold
Altruism	Selfish ≅ [< −6]	Normal ≅ [≥ −6 and ≤ 2]	Altruist ≅ [> 2]
Expertise	Beginner ≅ [< 4]	Normal ≅ [≥ 4 and ≤ 8]	Expert ≅ [> 8]

Table 3
Description of topic statistics for some of the proof-of-concept questions.

		ID question	48047128	48039340	48043536
	Topic classified	Title topic	Tools and setup	Programming in Java	Types files and data
		Body topic	Web basics	Input/output	Web basics
Posts stats	Metrics	Popularity	Normal	Normal	Mainstream
		AVG comments	3.28	3.44	3.28
		AVG time	295.96 h	176.17 h	252.95 h
		% URL	23,67%	14.48%	23.03%
		% Code	74.96%	80.76%	77.55%
Demography	Alstruism	Alstruists	55.03%	72.55%	40.16%
		Selfishness	6%	6.67%	6.03%
	Expertise	Experts	2.31%	2.9%	2.35%
		Beginners	22.96%	23.75%	22.53%
	SO level users	Bronze	12.35%	6.80%	18.93%
		Silver	23.7%	15.17%	35.7%
		Gold	46.22%	39.44%	60.76%

For example, such analysis enabled an extended classification of the users.

Another main focus of the study was the user dynamics, i.e. how often/much are Java developers prone to look for help in Stack Overflow and how many of these users show altruist and social profiles as opposed to egocentric behaviour. The corresponding graph profiled the users in terms of active participation and willingness to sustain continuous knowledge sharing. Noteworthy, categorization in Java is very important in case of answer time, in this particular metric, the difference by topic is very distant, most "quick" topic has an average of less than one hour and the most "slow" topic obtained an average of more than 12 days. Which denotes socializers and philanthropists of this community were focused on specific topics.

To the best of our knowledge, no previous work had addressed quality of service and user intrinsic motivation at this depth. As an immediate contribution, the proposed methodology was able to output valuable, non-trivial and up-to-date knowledge on a major social community of developers. Nevertheless, the concepts and techniques of analysis at the core of the methodology ensure broader application to other communities as well as similar Q&A platforms. Future work will be centred in evaluating alternative methods of topic modelling (i.e. extending modelling abilities of the method), exploiting graph mining for the enhancement of question routing, and experimentation over other communities (i.e. identifying and evaluating user motivation and Q&A flows of various nature).

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.

Acknowledgements

This study was supported by the Consellería de Educación, Universidades e Formación Profesional (Xunta de Galicia), Spain under the scope of the strategic funding of ED431C2018/55-GRC Competitive Reference Group, and the Portuguese Foundation for Science and Technology (FCT) under the scope of the strategic funding of UID/BIO/04469/2019 unit. SING group thanks CITI (Centro de Investigación, Transferencia e Innovación) from the University of Vigo for hosting its IT infrastructure.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.future.2019.12.021.

References

[1] StackExchange, 2019.
[2] T. Gruetze, R. Krestel, F. Naumann, Topic shifts in StackOverflow: Ask it like socrates, in: Int. Conf. Appl. Nat. Lang. to Inf. Syst., Springer, 2016, pp. 213–221, http://dx.doi.org/10.1007/978-3-319-41754-7.
[3] J. Sun, A. Vishnu, A. Chakrabarti, C. Siegel, S. Parthasarathy, ColdRoute: effective routing of cold questions in stack exchange sites, Data Min. Knowl. Discov. 32 (2018) 1339–1367, http://dx.doi.org/10.1007/s10618-018-0577-7.
[4] F. Calefato, F. Lanubile, N. Novielli, How to ask for technical help? Evidence-based guidelines for writing questions on Stack Overflow, Inf. Softw. Technol. 94 (2018) 186–207, http://dx.doi.org/10.1016/j.infsof.2017.10.009.
[5] S. Mumtaz, C. Rodriguez, B. Benatallah, Expert2Vec: Experts representation in community question answering for question routing, in: Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), Springer Verlag, 2019, pp. 213–229, http://dx.doi.org/10.1007/978-3-030-21290-2_14.
[6] Y. Igarashi, T. Altman, M. Funada, B. Kamiyama, Determining the popularity of design patterns used by programmers based on the analysis of questions and answers on stackoverflow.com social network, Computing (2014) 283–294, http://dx.doi.org/10.1201/b17011-30.
[7] M. Soliman, M. Galster, A.R. Salama, M. Riebisch, Architectural knowledge for technology decisions in developer communities: An exploratory study with StackOverflow, in: Proc. - 2016 13th Work. IEEE/IFIP Conf. Softw. Archit. WICSA 2016., 2016, pp. 128–133, http://dx.doi.org/10.1109/WICSA.2016.13.
[8] P. Yin, B. Deng, E. Chen, B. Vasilescu, G. Neubig, Learning to mine parallel natural language/source code corpora from stack overflow, in: 2018 IEEE/ACM 15th Int. Conf. Min. Softw. Repos., 2018, pp. 388–389, http://dx.doi.org/10.1145/3183440.3195021.
[9] K. Alreshedy, D. Dharmaretnam, D.M. German, V. Srinivasan, T.A. Gulliver, Predicting the programming language of questions and snippets of StackOverflow using natural language processing, 2018.
[10] C. Ragkhitwetsagul, J. Krinke, M. Paixao, G. Bianco, R. Oliveto, Toxic code snippets on stack overflow, IEEE Trans. Softw. Eng. PP (2019) 1, http://dx.doi.org/10.1109/TSE.2019.2900307.
[11] C. Greco, T. Haden, K. Damevski, StackInTheFlow: Behavior-driven recommendation system for stack overflow posts, in: Proc. 40th Int. Conf. Softw. Eng. Companion Proceeedings, 2018, pp. 5–8, http://dx.doi.org/10.1145/3183440.3183477.

[12] S. Fumin, X. Wang, S. Hailong, L. Xudong, Recommendflow: Use topic model to automatically recommend stack overflow Q & A in IDE, in: Collab. 2016 Collab. Comput. Networking, Appl. Work., Springer, 2017, pp. 521–526.

[13] L. Ponzanelli, G. Bavota, M. Di Penta, R. Oliveto, M. Lanza, Mining StackOverflow to turn the IDE into a self-confident programming prompter, in: 11th Work. Conf. Min. Softw. Repos., ACM, 2014, pp. 102–111, http://dx.doi.org/10.1145/2597073.2597077.

[14] C. Treude, M.P. Robillard, Augmenting API documentation with insights from stack overflow, in: 2016 IEEE/ACM 38th Int. Conf. Softw. Eng, 2016, pp. 392–403, http://dx.doi.org/10.1145/2884781.2884800.

[15] S. Gao, Z. Xing, Y. Ma, D. Ye, S.W. Lin, Enhancing knowledge sharing in stack overflow via automatic external web resources linking, in: Proc. IEEE Int. Conf. Eng. Complex Comput. Syst. ICECCS., 2018, pp. 90–99, http://dx.doi.org/10.1109/ICECCS.2017.30.

[16] A. Anderson, D. Huttenlocher, J. Kleinberg, J. Leskovec, Discovering value from community activity on focused question answering sites: a case study of Stack Overflow, in: Proc. ACM SIGKDD Conf. Knowl. Discov. Data Min., ACM, 2012, pp. 850–858.

[17] R. Slag, M. De Waard, A. Bacchelli, One-day flies on StackOverflow - Why the vast majority of StackOverflow users only posts once, in: IEEE Int. Work. Conf. Min. Softw. Repos., IEEE, 2015, pp. 458–461, http://dx.doi.org/10.1109/MSR.2015.63.

[18] S. Bird, E. Klein, E. Loper, Natural Language Processing with Python, first ed., O'Reilly Media, 2009, http://dx.doi.org/10.1162/coli_r_00022.

[19] M. Honnibal, M. Johnson, An improved non-monotonic transition system for dependency parsing, 2015, pp. 1373–1378, http://dx.doi.org/10.18653/v1/d15-1162.

[20] R. Rehurek, P. Sojka, Software framework for topic modelling with large corpora, in: Lr. Work. New Challenges NLP Fram., 2010, pp. 45–50.

[21] D.M. Blei, B.B. Edu, A.Y. Ng, A.S. Edu, M.I. Jordan, J.B. Edu, Latent Dirichlet allocation, J. Mach. Learn. Res. 3 (2003) 993–1022, http://dx.doi.org/10.1162/jmlr.2003.3.4-5.993.

[22] C. Sievert, K. Shirley, LDAvis: A method for visualizing and interpreting topics, 2015, pp. 63–70, http://dx.doi.org/10.3115/v1/w14-3110.

[23] J. Chuang, C.D. Manning, J. Heer, Termite: Visualization techniques for assessing textual topic models categories and subject descriptors, in: Proc. Int. Work. Conf. Adv. Vis. Interfaces. 2012.

[24] S. Kullback, R.A. Leibler, On information and sufficiency, Ann. Math. Stat. 22 (1951) 79–86.

[25] Y. Assenov, F. Ramírez, S.-E. Schelhorn, T. Lengauer, M. Albrecht, Computing topological parameters of biological networks, Bioinformatics 24 (2008) 282–284, http://dx.doi.org/10.1093/bioinformatics/btm554.

[26] L.C. Freeman, D. Roeder, R.R. Mulholland, Centrality in social networks: ii. experimental results, Soc. Netw. 2 (1979) 119–141, http://dx.doi.org/10.1016/0378-8733(79)90002-9.

[27] L.C. Freeman, Centrality in social networks conceptual clarification, Soc. Netw. 1 (1978) 215–239, http://dx.doi.org/10.1016/0378-8733(78)90021-7.

[28] L.C. Freeman, A set of measures of centrality based on betweenness, Sociometry 40 (1977) 35, http://dx.doi.org/10.2307/3033543.

[29] D.J. Watts, S.H. Strogatz, Collective dynamics of 'small-world' networks, Nature 393 (1998) 440–442, http://dx.doi.org/10.1038/30918.

[30] M. Bastian, S. Heymann, M. Jacomy, Gephi : An open source software for exploring and manipulating networks visualization and exploration of large graphs, 2019, www.aaai.org (accessed October 23, 2019).

[31] P. Shannon, A. Markiel, O. Ozier, N.S. Baliga, J.T. Wang, D. Ramage, N. Amin, B. Schwikowski, T. Ideker, Cytoscape: a software environment for integrated models of biomolecular interaction networks, Genome Res. 13 (2003) 2498–2504, http://dx.doi.org/10.1101/gr.1239303.

Guillermo Blanco González is Ph.D. student of Computer Science of the University of Vigo. He is currently developing advanced computational methods for modelling social dynamics, namely in biological ecosystems.

Roi Pérez López is a Master student of the Master in Computer Science of the University of Vigo. His main research interests include text mining, sentiment analysis, and topic modelling.

Florentino Fdez-Riverola is a Full Professor of the Department of Computer Science at the University of Vigo (Spain) and Coordinator of the New Generation Computer Systems group (SING, http://sing-group.org), which is dedicated to the research and development of cutting-edge computational methodologies and applications.

Anália Maria Garcia Lourenço is a faculty member of the Department of Computer Science and a researcher affiliated to the Biomedical Research Centre (CINBIO), at the University of Vigo and the Centre of Biological Engineering, at the University of Minho. Her main research interests include computational intelligence, bioinformatics and systems biology.