A Bird's Eye View on Social Network Sites and Requirements Engineering

Nazakat Ali[a] and Jang-Eui Hong[b]
Department of Computer Science, Chungbuk National University, Cheongju, Republic of Korea

Keywords: Social Network Sites, Requirements Engineering, Overview, User Requirements.

Abstract: Social network sites have become popular and their popularity is growing exponentially every day. From the requirements engineering point of view, social network sites have provided unprecedented opportunities for software development organizations to understand the requirements of unknown end-users. Using social network sites, end-users express their experiences, needs, or concerns about a particular system or a product. Such information can be useful for software developers to address the concerns of users quickly. To get an overview of how social network sites are helping requirements engineering and new research trends in this area, we have surveyed a large number of research papers. We found that social network sites can be a major source that can be used for requirements elicitation, requirements prioritization, and negotiation. We also found that the research in this domain is at its beginning stage, but it is rapidly growing with the passage of time.

1 INTRODUCTION

Social Network Sites (SNS) have become a major source for millions of users to share their daily activities, experiences, and opinions for a particular product through posting. As of year, 2019, 2.77 billion users around the globe use SNS (Statista 2019). As a result, a huge amount of real-time and highly diverse data is produced per day, which brought a revolution in many research fields such as in the data science field. In data science, this data can be used to predict the political affiliations, forecasting stock market fluctuations, and marketing trends, etc. (Conver 2011). Being a communication channel, SNS have been used in various software development activities such as in Requirements Engineering (RE). In RE, SNS have proved their support and improved several process activities such as requirements elicitation, negotiation, prioritization (kukreja 2012 and Seyff 2015), and identification and prioritization of stakeholders (Lim et al. 2010). RE is a critical activity of the software development lifecycle and user participation in RE can lead to success in software projects (Hofman et al. 2001). Therefore, it seems most suitable to use SNS in order to conduct various activities of RE due to the presence of a huge population on it.

The purpose of this study is to present a bird's eye view survey of how SNS benefit RE activities. More specifically, we are formulating the following research questions to accomplish our goal.

RQI: What is the current state-of-the-practice of the SNS (Facebook, Twitter, etc.) as a platform to conduct RE activities?
RQ2: What RE activities are supported by SNS?
RQ3: Is SNS improving the state-of-the-art of RE?
RQ4: What challenges requirements engineers are face while using SNS to conduct RE activities?

Based on these formulated research questions, we have conducted a literature review to summarize the existing literature in which SNS are considered as a platform to conduct RE activities. The literature review is conducted by splitting it through preparation, data collection, and data analysis phase. In preparation, a search string was prepared to search the literature for answering formulated questions, then searching was performed over a number of databases such as Scopus, IEEEXplore, Springer Link, Google Scholar, ACM Digital Library, and Science Direct. All the identified research articles

[a] https://orcid.org/0000-0002-3875-812X
[b] https://orcid.org/0000-0001-9786-7732

Ali, N. and Hong, J.
A Bird's Eye View on Social Network Sites and Requirements Engineering.
DOI: 10.5220/0008117303470354
In *Proceedings of the 14th International Conference on Software Technologies (ICSOFT 2019)*, pages 347-354
ISBN: 978-989-758-379-7

were assessed by means of their title and abstract. Along with this, references and citations of specific articles were reviewed to explore more relevant research articles of our focus. Our intention is not to conduct an extensive literature review of the targeted field - rather we are presenting a snapshot of the state-of-the-art by following some guidelines (study selection, formulating research questions, information synthesis, and result reporting) of Kitchenham (Kitchenham 2004) to conduct systematic literature in our targeted field.

The major contributions of this study are as follows:

- We are presenting an overview of the state-of-the-art on how SNS supports existing RE activities. Precisely, we are focusing on how requirements elicitation, requirements analysis, and requirements management is supported by utilizing SNS.
- Since machine learning is utilized to carry out all these mentioned RE activities, therefore, we are offering an overview on how machine learning algorithms (classification, clustering, regression, etc.) are utilized to get full advantage of SNS data.
- Furthermore, we analyze existing literature to uncover research trends and future research directions on the use of SNS for conducting RE activities.

The rest of this paper is organized as follows. Section 2 provides necessary background information on SNS and RE is required to understand the rest of the paper. Section 3 presents an overview of RE activities where SNS is used as a platform to conduct RE. In section 4, we summarize our major findings of this research and answer the formulated questions while section 5 concludes this paper.

2 SOCIAL NETWORK SITES AND REQUIREMENTS ENGINEERING

The fundamental goal of a software system is to meet the requirements of its users by offering functionalities that can satisfy and fulfill the expectations of its users. This goal is accomplished by applying several engineering techniques. RE is an activity through which user requirements are identified for a specific domain in a systematic manner to understand what features a specific product should have in order to satisfy its users' expectations.

The success of a software project mainly depends on its RE process because requirements are the determining factor of software quality. The empirical evidence has shown that errors in the requirements are most common in the software development lifecycle and it is considered as most time- consuming and expensive task to correct in later stages (Wohlin, 2005). According to a research report (Standish, 2014), 52.7% of projects met with challenges and problems and cost 189% of their total estimated budget. The report shows that only 16.1% of all US projects completed on-budget, on-time with all planned features, while 31.1% of projects were never completed. The investigation report identified poor requirements as the main source of problems, along with other factors such as unclear objectives and low user involvement. Likewise, another survey (European 1996) which was carried out with 3800 organizations from 17 countries in Europe, the findings show that half (50%) of the problems were due to requirements specification and rest half (50%) problems emerged due to requirements management.

3 EXISTING CONTRIBUTIONS

In this section, we will summarize the existing literature which has considered SNS as a platform to conduct RE activities. A comprehensive inclusion and exclusion criteria were followed to select the literature. As a whole, Kitchenham et al. guidelines (Kitchenham, 2004) were partially followed to conduct this survey.

The success of any software project majorly depends on the accurate identification of stakeholders' expectations and requirements for their anticipated system. Requirements elicitation through the manual process is costly in terms of resource and efforts. Therefore, SNS provided a platform to conduct requirements elicitation by mining microblogs e.g., social media, app store reviews, and requirement documents. The mining task is carried out by applying various techniques such as Natural Language Processing (NLP), text mining and Machine Learning (ML) (Ali, 2016). In current practice, popular SNS such as Twitter, Facebook, and Snapchat are used to extract user requirements by applying ML techniques. User comments, tweets, or reviews are not structured documents, they contain useful information along with noise which makes difficult to extract user requirements with manual annotations. Therefore, ML techniques are most favorable to identify user requirements automatically

or semi-automatically by reducing efforts, cost, and time significantly.

3.1 RE, ML and Social Network Sites

Guzman et al. (Guzman, 2016) have proposed an approach that uses Twitter as a platform to elicit user requirements. Their exploratory study investigated the content of 10,986,494 tweets about 30 software application. The authors have categorized tweets into more 22 categories including feature shortcomings, bug reports, feature request, and feature strength, etc. The authors have applied Support Vector Machine (SVM) and C4.5 ML classifiers to classify the tweets automatically. Both classifiers had a similar performance to classify tweets related to non-technical stakeholders. However, SVM performed well to classify tweets related to technical stakeholders and the general public.

Guzman et al. (Guzman, 2017(a)) presented a similar study for classifying, ranking and grouping tweets for software evolution. The authors used Twitter as a platform to elicit user requirements. The proposed approach classifies tweets into two categories including improvement request and others. The authors used 68,108 tweets for their experiment. Naïve Bayes (NB) and Random Forest (RF) classifiers were used to classifying the tweets. In summary, the proposed approach was able to classify tweets automatically into improvement request and other categories with F-measure of 0.79.

Williams et al. (Williams, 2017(a)) proposed an approach which also used Twitter as a platform to elicit user requirements. The authors first analyzed 4000 tweets of 10 various software systems to classify tweets into bug reports, user requirements and miscellaneous and spam. Then, the authors used NB and SVM to classify the tweets automatically. The authors used three classification features (bag-of-words, sentiment and stemming) for their tweets classification. The study claims that the sentiment feature did not affect classification results. The authors used 188,737 unique tweets for their experimentation.

Xiao et al. (Xiao, 2015) have used popular SNS such as StackOverflow, a popular Q & A software community site, where programmers and software engineers share their knowledge and experience. The collected data from StackOverflow was used to extract requirements requests to help software developers. In their study, SVM classifier was used to test the data. The authors extracted tagged questions along with related information to elicit requirement requests for a relevant software application. Initially, three ML algorithms including pure SVM, linguistics rules and SVM with dictionary were selected for experiments and comparison. SVM with requirement dictionary is the combination of a dictionary (keyword sets) and SVM to enhance the performance of classification. It is concluded from the experimental result that SVM, with precision 69%, recall 75% and F-measure 72%, performed better than traditional linguistic rules. SVM with requirement dictionary improved performance with precision 72%, recall 77% and F-measure 74.4%.

Prasetyo et al. (Prasetyo, 2012) performed a preliminary study to investigate the feasibility of automatic classification of microblogs into relevant and irrelevant categories. All those tweets that could help software developers to understand user needs were categorized as relevant tweets while rest were categorized as irrelevant tweets. The authors used 300 tweets of [32] comprising either one of the following 9 hashtags including #csharp, #scrum, #javascript, #opensource #.net, #jquery, #testing, #azure, and #java, and tweets were classified into 10 categories including news, commercials, tools and code, events, personal, tips, opinions, jobs, Q &A, and miscellaneous. The aim of this study was to classify tweets automatically into relevant and irrelevant. Therefore, the authors have re-labeled data set into relevant and irrelevant categories. If a tweet was potentially interesting to a software developer to develop a target system, otherwise they labeled it as irrelevant. The result shows that 41% of tweets were relevant while 53% were irrelevant. For the automatic classification of tweets, the authors used SVM classifier.

Guzman et al. (Guzman et al., 2017(b)) used Twitter to elicit user requirements. This study reports on the usage characteristics, contents, and automatic classification potential of user tweets about RE and software evolution. The authors explored a dataset containing 10,986,495 tweets about 30 various software applications by using descriptive statistics, content analysis, lexical sentiment analysis, and ML. The authors have used NB, multinomial NB, SVM, J48, and RF to classify a huge amount of user tweets according to the relevance to different stakeholders and user type. The result shows that the classifiers produced promising results in some cases. The automated relevance classification was possible with an F-score ranging from 77% to 52%, while the identification of tweets tweeted by bots got a promising F-score of 84%. This study has performed a fine granular evaluation of user feedback from Twitter and its comparison with App store reviews.

Deshpande et al. (Deshpande et al., 2018) proposed a study that assessed user feedback from Twitter in terms of timing as well as content and compared with the App store review. This study employed various text analysis and NLP methods such as semantic analysis to analyze tweets and app store reviews. The authors have used SVM and NB for their classification. The analysis of topics showed that 72% of tweets were functional requirements which discussed the behavioral aspects of mobile apps while in the App store it was 80%.

Singer et al. (Singer et al., 2014)) surveyed 271 GitHub users and interviewed 27 software developers to know about the importance of Twitter in software development. Authors found that developers often rely on various online resources, such as Twitter to keep themselves informed about their systems. The authors also found that Twitter helps software developers to stay aware of industry changes, technology changes and for learning.

Williams et al. (Williams et al., 2017(b)) presented a preliminary study aiming to detect and interpret emotions present in software related tweets. The preliminary study was conducted using a dataset of 1000 tweets taken from tweets of a broad range of software applications. At first, the tweets were manually classified by two annotators with 5 years of programming into two levels of abstractions, including its general emotional polarity such as positive, negative and neutral, and the specific emotions it conveys. After manual classification dataset was trained and applied SVM and NB to extract sentiment automatically. The authors have seen that emotions were closely related to specific events such as product release etc. The results also show that tweets which carry negative sentiment usually report bugs, report feature shortcoming, etc.

Seyff et al. (Seyff et al., 2018) propose a crowd and sustainability-focused platform for semi-automated requirements elicitation, negotiation, and analysis. This platform enables a diverse and distributed group of stakeholders or users (crowd) to communicate and negotiate their requirements. The proposed conceptual solution includes three components: 1) CrowdFeed enables users to give their feedback regarding the software service or products they use, 2) Requirements and Sustainability Service classifies, clusters, and analyze the feedback received from the crowd, 3) Requirements and Sustainability Integrator supports the visualization and assessment of effects on sustainability.

3.2 RE and Social Network Sites

Lee et al. (Lee et al., 2011) used both Twitter and Facebook for requirements elicitation and prioritization. The proposed approach has three phases including preparing phase, gathering phase and refining phase. The preparing phase describes the preparation for requirements elicitation through SNS. In the preparation phase, a keyword-based method is used to approach a large number of user opinions. The authors first selected domain related keywords e.g. smartphone app, smartphone OS, smartphone design, etc. The frequency of selected domain-related keywords was recorded to select those keywords which are most discussed on SNS. In the gathering phase, the SNS and its users were approached, and their opinions were collected. This phase was further divided into two parts according to types of access to the subject and the direction of approach: opinion-based access and community-based access. Opinion-based access targeted individual users, not SNS communities while in community-based access diverse communities were targeted and their opinions were collected.

Seyff et al. (Seyff et al., 2015) proposed an approach for RE that employs Facebook for elicitation, prioritization, and negotiation of requirements. The authors have conducted three exploratory studies using Facebook to see whether the potential of popular SNS (i.e. Facebook) can allow and support end users to participate in requirements elicitation, prioritization, and negotiation. This study is inspired by EasyWinWin (Gruenbacher, 2000) and authors foresaw that their proposed lightweight and end-user focused RE approach had many potential applications. The authors particularly see its relevance with new software paradigms such as cloud computing, mobile computing, and ecosystems, where potential end users are not within instant reach and support provided by traditional RE methods is insufficient. The authors claim that their approach can provide a new channel for eliciting innovative ideas and needs as well as getting feedback from stakeholders who are not directly reachable by software development organizations or software development teams. Furthermore, authors see its applicability within traditional software projects as an additional means for engaging end users.

Ali et al. (Ali et al., 2017) considered both Twitter and Facebook to evolve systems. The authors proposed a cyclic process that elicits user requirements continuously from SNS. After data extraction, the corpus is gone through a comprehensive NLP phase to elicit user requirements.

In this approach, bigrams and trigrams were exacted to get the related keywords. Later these bigrams and trigrams were interpreted as user requirements. Furthermore, the authors used the ontology-based similarity algorithm to detect redundant keywords. After requirements identification, a feature model was manually built. After every iteration, the feature model was updated, and the system was reconfigured to adapt to the changes. In another study (Ali et al., 2016) authors have elicited domain requirements from SNS to support software product line evolution. This study has analyzed Facebook, Twitter, and LinkedIn and selected Twitter and Facebook as a platform to elicit user requirements.

Based on the win-win (Gruenbacher, 2000) negotiation model, Kukreja et al. (Kukreja, 2012) developed a social network site such as "Winbook", which is based on the social networking paradigm. Winbook is similar to Facebook, but its contents are organized using color-coded labels similar to Gmail. It also resembles Excel because it has the ability to prioritize requirements by performing a number of sensitivity analyses with respect to business goals. This study is able to elicit, negotiate and prioritize user requirements.

Romasha et al. (Romasha et al., 2018) have conducted a survey to know how SNS empower RE. In this questionnaire-based survey, among respondents, 41% were software developers, 31 were team leaders, and 10% of respondents were a business analyst. As per the survey result, majority respondents agreed that RE is a communication activity and 73.8 % respondent expressed that they face communication problem to gather requirements. Among respondents, 57% of respondents reported that they used Facebook to elicit user requirements and solve the communication problem.

Maalej et al. (Maalej et al., 2015)] have discussed the usage of SNS and app stores to elicit user requirements. The authors discussed how software development organizations can use user feedback to identify, prioritize, and manage requirements. The authors have seen three directions in practice. First, tools for feedback analytics would help in order to deal with a large number of user opinions by classifying, summarizing, and filtering them. Second, automatically collected opinions, logs, and interaction traces could help software developers to understand user feedback and improve feedback quality. Third, once the feedback is collected and analyzed, the question arises how can software development organizations use this information and integrate it into their software development process to decide when to release new updates. What feature or requirements should be added, deleted, or changed?

Groen et al. (Groen et al., 2017) have proposed CrowdRE approach to elicit user requirements from the crowd. The authors considered the crowd as a sender of the feedback and software development companies as a receiver. The authors dived user feedback into two categories: pull feedback and push feedback. Push feedback is when software development organizations explicitly ask the crowd for feedback while the push feedback is when the user in the crowd start a discussion. In both cases, user feedback was utilized to elicit user requirements. Additionally, the authors have listed a number of challenges as well including crowd motivation, privacy and personalization, and feedback analyzation issues.

Mughal et al. (Mughal et al., 2018) have proposed a network-based process using Facebook to reduce the in-group biasness during RE. The proposed approach combined hybrid centrality measure and power, legitimacy, and urgency technique. This study has focused on stakeholder identification and prioritization during RE. The main strength of this article is that it has shown better performance than partner biasness issue through hybrid centrality measure and the power, legitimacy, urgency and urgency model, as claimed. The feasibility of the proposed social network-based process for identification and prioritization of requirements of stakeholders is shown through a controlled experiment conducted on an example set of 40 stakeholders.

Kanchev et al. (Kanchev et al., 2017) have proposed a methodology called Canary that generates a requirements-oriented view of online discussions. The methodology is a conceptual model that brings requirements-relevant and argumentation-relevant aspects of online discussions together. The authors have claimed that it was feasible to convert online discussions into instances Canary model by crowdsourcing annotations of the discussions.

4 RESULTS AND DISCUSSION

Our study focuses on a specific area and points to a number of trends that we will explain in this section. Overall, the goal of this study is to provide a state-of-the-art of use of SNS to conduct various activities of RE for giving an overview to domain experts and act as a foundation to this field for researchers. Furthermore, the results we provide here are derived from the literature review we conducted. Our survey

Table 1: Contributions and RE activities supported by SNS.

N	Contributions	Publication Year	Objective/Purpose	ML Tasks	RE Activities Support
1	Guzman et al. (+)	2016	Categorization of tweets into 22 categories including bug reports, feature shortcoming, feature demand etc.	Classification	Elicitation
2	Guzman et al. (+)	2017(a)	Classification of tweets into two categories: improvement request and others.	Classification	Elicitation
3	Williams et al. (+)	2017(a)	Classification of tweets into bug reports, user requirements, and miscellaneous	Classification Wordcloud	Elicitation
4	Xiao et al (-)	2015	Extract requirement requests	Classification	Elicitation
5	Lee et al. (+)	2011	Requirements elicitation and prioritization	-	Elicitation Prioritization
6	Seyff et al. (-)	2015	Requirements elicitation, negotiation, and prioritization using SNS	-	Elicitation Prioritization Negotiation
7	Ali et al. (*)	2017	Eliciting user requirements from SNS to evolve software product lines	-	Elicitation Prioritization
8	Prasetyo et al. (-)	2012	To know the relevance of tweets with software systems	Classification	Elicitation
9	Guzman et al. (+)	2017(b)	searching for relevant information on software applications within the vast stream of tweets	Classification	Elicitation Prioritization
10	Deshpande et al. (*)	2018	Comparison between App store reviews and Tweeter messages for requirements	Classification	Elicitation
11	Kukreja et al. (+)	2012	Developing Winbook	-	Elicitation Prioritization Negotiation
12	Romasha et al. (*)	2018	Conducting a survey that to know how SNS empower RE	-	Elicitation Negotiation
13	Maalej et al. (*)	2016	The authors discussed how software development organizations can use user feedback to identify, prioritize, and manage requirements	-	Elicitation Prioritization Negotiation
14	Leif et al. (*)	2014	Surveying and interviewing software developers to know about the importance of Twitter in software development.	-	-
15	Williams et al. (-)	2017(b)	presenting a preliminary study aiming to detect and interpret emotions present in software related tweets.	Sentiment	Elicitation
16	Groen et al. (*)	2017	Proposing CrowdRE approach to elicit user requirements from crowd.	-	Elicitation
17	Mughal et al. (+)	2018	proposing a network-based process using Facebook to reduce the in-group biasness during RE	-	Stakeholders prioritization in requirements elicitation process
20	Ali et al. (-)	2018	Introducing the idea of a platform for crowd-focused requirements engineering that supports the evolution towards sustainability	Multi-label classification	Intended to elicit requirements
21	Kanchev et al. (+)	2017	extracting and querying requirements-related information in online discussions (reddit)	-	Elicitation
22	Seyff et al. (-)	2018	sustainability-focused platform for semi-automated requirements elicitation, negotiation, and analysis	Classification	Elicitation, Negotiation

Legends: (+) improves the state of the art; (*) no information how it connects with state of the art; (-) comparable with state of the art;

is not a fully systematic literature review (as mentioned in section 1) and our conclusions and results can be revised or extended by future studies in this domain. Table 1 provides answers to **RQ1** ("What are the current state-of-the-practice that use SNS as a platform to conduct RE activities?") and **RQ2** ("What RE activities are supported by SNS?"). Twenty-one papers (Contributions column in Table 1) were thoroughly examined to know the state-of-the-art which employs SNS to conduct RE activities. We have seen that; the majority of studies were initial proposals with a little academic literature and no industrial applications in real software projects. ML tasks and RE support Activities columns, as mentioned in Table 1, answer our **RQ2**. The contributions column in Table I also provides a partial response to **RQ 3** ("Is SNS improving the state-of-the-art of RE?"). The answer to **RQ3** appears to be at its beginning, given the prevalent lack of comparison with the state of the art as can be seen in Table 1. The

RQ3 was answered by comparing proposed approaches, that consider SNS for RE, with traditional state-of-the-art. Some additional information is also given in Table 1 i.e. what kind of ML models were used in order to elicit user requirements from the corpus. We have seen that NB and SVM are mostly used for classification.

The SNS contains a huge amount of data including user expectations and experiences. Software developers could use this rich information to understand the needs, experiences, and sentiment of their product users. Mining SNS especially user opinions, may yield useful information for the software development organization to update their products. However, extracting user requirements out of huge data is also a challenging task. The opinions are usually expressed without considering grammar format which creates problems lately for corpus processing. Therefore, we have seen that the collected data was unstructured. Software developers rely on user feedback for requirements elicitation, negotiation, and prioritization. However, the trustworthiness of comments, tweets, or, in general, feedback is still a big challenge for the software development community. Groen et al. (Groen et al. 2017) have mentioned a number of challenges in this domain. User motivation is one of the big challenges to get feedback. User privacy and personalization are other issues which RE community faces while using SNS as a platform to elicit user requirements. These observations answer **RQ4** ("**RQ4:** What are the challenges requirements engineers face while using SNS to conduct RE activities?").

Table 1 did not mention the NLP techniques applied in our surveyed papers. It is obvious that NLP techniques are widely used by the majority of researchers in their research that tackles the application of SNS for RE. This is not astonishing and even more common in data science. We observed that NLP was used in the preprocessing phase in order to bring data into the desired format that can be consumed by ML algorithms. Majority of the articles we have gone through in our survey have the opinion that a large amount of imprecise data produced by SNS users can bring about enormous benefits to software development organizations when processed by applying ML algorithms. The majority of papers we surveyed on the topic of SNS and RE have to do with either elicitation or prioritization activities of RE and a little literature talked about negotiation activity as well. Figure 1 shows the result of our investigation.

As mentioned above, the research in the domain of RE and SNS is at its beginning but the research trend is growing rapidly with the passage of time.

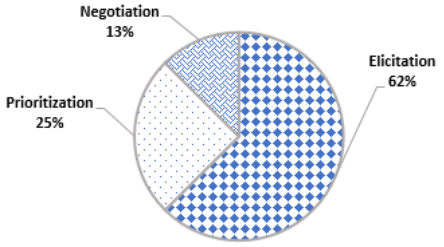

Figure 1: RE activities supported by SNS in our study.

Figure 2 shows the number of publications with the passage of time which clearly shows the ongoing research trend on SNS and RE domain. Figure 2 also shows the number of publications that used ML techniques to support RE activities.

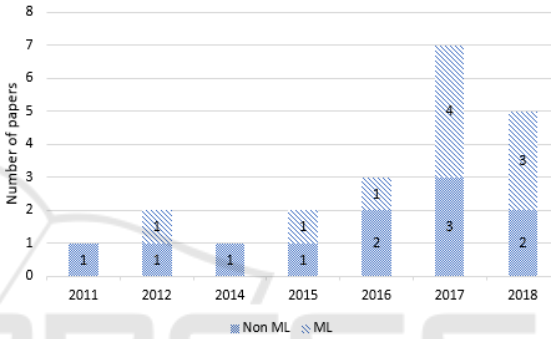

Figure 2: Number of publications in SNS and RE domain.

5 CONCLUSIONS

Through our bird's eye view on SNS and RE, we have seen that in the past few years a research trend has started to bring these two worlds together. The investigation shows that studies in this domain are at the beginning stage. Although, the academicians have started to explore the potentials of SNS in order to use it to conduct RE activities.

We have conducted a literature survey to know how SNS is being used to conduct RE activities. We formulated four types of research questions (Section I) and extracted data according to answer these questions. RQ1 revealed the ongoing research that attempted to bring together SNS and RE worlds. RQ2 exposed that what kind of RE activities are supported by SNS and what are the ML algorithms are used to extract the required information from the SNS data. In RQ3, we have seen that research in this domain is in its early stages and need more reliable approaches that will benefit software projects in the real world. Finally, RQ4 answered that what challenges are faced by the research community while bringing two worlds together.

ACKNOWLEDGMENTS

This research was supported by Next-Generation Information Computing Development Program through the NRF of Korea funded by the Ministry of Science, ICT (NRF-2014M3C4A7030505, NRF-2017M3C4A7066479).

REFERENCES

Ali, N., Kim, S. and Hong, J.E., 2016. Listen closely, respond quickly: Enhancing conformity of SPL domain requirements through SNS. International Conference on Information Science and Communications Technologies (ICISCT*)*, pp. 1-5.

Ali, N. and Hong, J.E., 2017. Creating adaptive software architecture dynamically for recurring new requirements. International Conference on Open Source Systems & Technologies (ICOSST) (pp. 67-72.

Conover et al., 2011, O. Predicting the political alignment of twitter users. In *2011 IEEE third international conference on privacy, security, risk and trust and 2011 IEEE third international conference on social computing*, pp. 192-199.

Deshpande, G. and Rokne, J., 2018. User Feedback from Tweets vs App Store Reviews: An Exploratory Study of Frequency, Timing and Content. 5th International Workshop on Artificial Intelligence for Requirements Engineering (AIRE) (pp. 15-21.

European Software Institute., 1996. European user survey analysis. Report USV_EUR 2.1, ESPITI project.

Guzman, E., Alkadhi, R. and Seyff, N., 2016. A needle in a haystack: What do twitter users say about software? In IEEE 24th International Requirements Engineering Conference (RE), pp. 96-105.

Guzman, E., Ibrahim, M. and Glinz, M., 2017(a). A little bird told me: Mining tweets for requirements and software evolution. IEEE 25th International Requirements Engineering Conference (RE), pp. 11-20.

Guzman, E., Alkadhi, R. and Seyff, N., 2017(b). An exploratory study of Twitter messages about software applications. RE, 22(3), pp.387-412.

Gruenbacher P., 2000., Collaborative requirements negotiation with EasyWinWin," Proceedings 11th International Workshop on Database and Expert Systems Applications, pp. 954-958.

Groen, E.C.,et al., 2017. The crowd in requirements engineering: The landscape and challenges. *IEEE software*, *34*(2), pp.44-52.

Hofmann, H.F. and Lehner, F., 2001. Requirements engineering as a success factor in software projects. *IEEE software*, (4), pp.58-66.

Kukreja, N., 2012,. Winbook: a social networking-based framework for collaborative requirements elicitation and WinWin negotiations. In Proceedings of the 34th International Conference on Software Engineering, pp. 1610-1612.

Kitchenham, B., 2004. Procedures for performing systematic reviews. *Keele, UK, Keele University*, pp.1-26.

Kanchev, G.M., Murukannaiah, P.K., Chopra, A.K. and Sawyer, P., 2017. Canary: Extracting requirements-related information from online discussions. 25th International Requirements Engineering Conference (*RE)*., pp. 31-40.

Lim, S.L., Quercia, D. and Finkelstein, A., 2010. StakeNet: using social networks to analyse the stakeholders of large-scale software projects. In *Proceedings of the 32Nd* International Conference on Software Engineering-Volume 1 (pp. 295-304).

Lee, Y.K., Kim, N.H., Kim, D., Lee, D.H. and In, H.P., 2011. Customer Requirements Elicitation based on Social Network Service. KSII Transactions on Internet & Information Systems, 5(10).

Maalej, W., Nayebi, M., Johann, T. and Ruhe, G., 2015. Toward data-driven requirements engineering. IEEE Software, 33(1), pp.48-54.

Mughal, S., Abbas, A., Ahmad, N. and Khan, S.U., 2018. A Social Network Based Process to Minimize In-Group Biasedness During Requirement Engineering. *IEEE Access*, *6*, pp.66870-66885.

Number of social media users worldwide from 2010 to 2021 (in billions https://www.statista.com/statistics/278414/number-of-worldwide-social-network-users/ accessed January 23, 2019.

Prasetyo, et al., 2012, September. Automatic classification of software related microblogs. 28th IEEE International Conference on Software Maintenance (ICSM) (pp. 596-599

Romasha K, et al., 2018., Social Networking Sites Empower Requirement Engineering" 31st IBIMA Conference. pp. 25-26 Milan, Italy.

Singer, L., Figueira Filho, F. and Storey, M.A., 2014. Software engineering at the speed of light: how developers stay current using twitter. In Pros of the 36th International Conference on Software Engineering (pp. 211-221

Seyff, N., Betz, S., et al. 2018, August. Crowd-focused semi-automated requirements engineering for evolution towards sustainability. In *2018 IEEE 26th International Requirements Engineering Conference (RE)*, pp. 370-375.

Seyff, N., Todoran, I., Caluser, K., Singer, L. and Glinz, M., 2015. Using popular social network sites to support requirements elicitation, prioritization and negotiation. *Journal of Internet Services and Applications*, *6*(1), p.7.

Standish. C Group., 2014. The standish group report.

Wohlin, C., 2005. Engineering and managing software requirements. Springer Science & Business Media.

Williams, G. and Mahmoud, A., 2017(a). Mining Twitter feeds for software user requirements. IEEE 25th International Requirements Engineering Conference (RE)(pp. 1-10.

Williams G and A. Mahmoud., 2017(b) Analyzing, Classifying, and Interpreting Emotions in Software Users' Tweets, 2nd International Workshop on Emotion Awareness in Software Engineering (SEmotion), pp. 2-7

Xiao, M., Yin, G., Wang, T., Yang, C. and Chen, M., 2015. Requirement acquisition from social q&a sites. In Requirements Engineering in the Big Data Era (pp. 64-74.