

CREeLS: Crowdsourcing based Requirements Elicitation for eLearning Systems

Nancy M. Rizk¹, Mervat H. Gheith², Ahmed M. Zaki³, Eman S. Nasr⁴
Business Information Systems Department, College of Management and Technology
Arab Academy for Science, Technology, and Maritime Transport (AASTMT), Egypt¹
Faculty of Graduate Studies for Statistical Research, Cairo University, Egypt²
Independent Researchers, Egypt^{3,4}

Abstract—Crowdsourcing is the process of having a task performed by the crowd. Because of the Web evolution, recently crowdsourcing is being used in the field of Requirements Engineering to help in simplifying its activities. Among the information systems that were highly affected by the Web evolution are the eLearning Systems (eLS). eLS has special characteristics, such as the large number and diversity of users who could be geographically dispersed. To the best of our knowledge, there is little evidence that a crowdsourcing based requirements elicitation approach especially tailored for eLS that addresses their special characteristics exists. In this paper we attempt to fill in this gap. We present Crowdsourcing based Requirements Elicitation for eLS (CREeLS), which is made up of a framework of the necessary elements of crowdsourcing suggesting specific tools for each element, and a phased approach to implement the framework. We evaluated our approach through analyzing real-life users' reviews and extracted keywords that represent users' requirements by using topic modeling techniques. The reached results were then evaluated by manual text reviewing and the extracted features were found to be coherent. CREeLS has 0.66 *precision* and 0.79 *recall*. Hence we contend that CREeLS can help requirements engineers of eLS to analyze users' opinions and identify the most common users' requirements for better software evolution.

Keywords—Requirements engineering; requirements elicitation; crowdsourcing; eLearning systems

I. INTRODUCTION

Crowdsourcing is known to be the process of obtaining the needed services by outsourcing them to the crowd, which can be defined as a group of people with a common interest [1]. Crowdsourcing as a term was coined by Jeff Howe in Wired in June 2006. In his article "The Rise of Crowdsourcing" [2], he described how small businesses are getting successful from using "the power of the crowd", rather than the traditional professional ways, to cut their costs. He also proposed the use of crowdsourcing in research and development departments to get new ideas for new products or features from the crowd. Howe then wrote an article titled "Crowdsourcing: A Definition" [3] to make the term clearer and not to confuse it with other terms. He mentioned that crowdsourcing can be used at any time when needed in an organization.

The use of "the power of the crowd" to achieve specific tasks is gaining more and more ground every day. Because of the Web evolution, recently crowdsourcing is being used in the field of requirements engineering to help in simplifying the

activity of requirements elicitation, which usually involves various stakeholders, e.g. [4] [5] [6] [7] [8]. According to Sommerville [8], requirements engineering activities are: requirements elicitation, requirements analysis, requirements specification and requirements validation. This paper will focus on the requirements elicitation activity only.

Requirements elicitation is an early software development activity within the requirements engineering phase. It is concerned with understanding and learning stakeholders' needs [9]. It is a very important activity for the success of a software development project, where detecting errors at the early stages of development can save money and time [10]. According to the Standish Group CHAOS Report [10], users' involvement is one of the critical success factors in any software development project and this is usually performed in the requirements elicitation activity. The narrow concept of stakeholders, the limited involvement of users with knowledge in requirements prioritization, and the bias of a requirements engineer who focus on certain types of requirements, in addition to the geographically dispersed stakeholders, and the uncertain technical and social environment are among the limitations and threats of the traditional requirements elicitation approaches [11] [12] [13]. Crowd-based requirements engineering, which was coined as a term by Groen et al., is a highly interactive approach; it can get user requirements in less time, helps in getting new ideas for software evolution, and has the potential to increase the quality of requirements elicitation [4].

Among the information systems that were highly affected by the Web evolution are the eLearning Systems (eLS). eLearning is well known to be the use of technology in the delivery of education, where in some cases the learning resources are accessed online anywhere and anytime [15], [16]. eLearning has different tools, types, and information systems, which are discussed and compared in [17]. eLS has two main types of management systems; Learning Management Systems (LMS) and Learning Content Management Systems (LCMS). LMSs are concerned with the administrative process of learning, such as scheduling, testing, billing and registering learners, e.g. Moodle, and Blackboard [18]. LCMSs combine the administrative processes of LMS with the authoring and content creation dimensions [19] [20]. Hence eLS can be defined to be the systems that are concerned with the administering or content authoring tools to help students, instructors and management in the learning process. eLS can

be used in an educational context or a corporate training context.

Among the characteristics of eLS are the large number and diversity of eLearning users in terms of background, geographical locations, and culture. Also, the high interactivity nature of the learning process, leads to an on-going demand of requirements that should be fulfilled for better improvement of the learning process and satisfaction of stakeholders. Limitations of the traditional requirements elicitation approaches are also exists when requirements elicitation are performed on eLS [21], [13]. Hence, we propose crowdsourcing too to be used in the eLearning context to handle the eLS characteristics and serve in the requirements elicitation activity. To the best of our knowledge, there is little evidence that a crowdsourcing based requirements elicitation approach especially tailored for eLS, that addresses their special characteristics exists. In this paper we attempt to fill in this gap to increase the quality of eLS' requirements elicitation and be able to get user requirements in less time, or get new ideas for software evolution by reaching greater number of stakeholders no matter their location or culture. We claim that crowdsourcing can use "the power of the crowd" through the power of Web 2.0 technologies to better elicit the stakeholders' requirements for eLS.

In this paper we present Crowdsourcing based Requirements Elicitation for eLS (CREeLS), which is made up of a general framework of the necessary elements of crowdsourcing suggesting specific tools for each element, and a phased approach to implement the framework in the requirements elicitation activity for eLS. We conducted an experimental study to evaluate the validity of our proposed approach. In addition, manual reviewing of user's requirements was used for the evaluation of the experimental study. CREeLS has 0.66 precision and 0.79 recall. Hence we contend that CREeLS can help requirements engineers of eLS to analyze users' opinions and identify the most common users' requirements for better software evolution.

The rest of this paper is organized as follows. Section 2 provides the literature survey of requirements elicitation for eLS, crowdsourcing-based requirements engineering, and the use of crowdsourcing in eLS as published in the literature. Section 3 presents CREeLS, our proposed framework and proposed approach, to attempt to fill in the identified gap in the literature. Section 4 explains and discusses the experimental study conducted using CREeLS. Section 5 discusses the evaluation of the experimental study, its results and limitations. Finally, Section 6 gives the conclusion and future work.

II. LITERATURE SURVEY

This section gives the literature survey; it was conducted for three main fields of study. It first presents a brief survey about available publications of requirements elicitation for eLS. Second, it surveys the different crowdsourcing-based tools and approaches published in the requirements engineering literature. Third, and finally, it presents how crowdsourcing is used in general in the eLS literature.

A. Requirements Elicitation for eLS

In our attempt to make a survey to study the limitations of the traditional requirements elicitation approaches for eLS; we only found very little publications that address requirements elicitation for eLS in particular. AlKhuder and AlAli [21] presented the importance of requirements elicitation activity for the eLS development, mentioned some of eLS characteristics that challenge the requirements elicitation activity e.g. the on-going demand of requirements of learners, and the variability of stakeholders. They also proposed some eLS requirements for different aspects of the system as an outcome of requirements elicitation activity. On the other hands, the authors were not clearly revealing the source or basis of eLS requirements presented in their paper.

Abdul Rahman and Sahibuddin [13] discussed the challenges of requirements engineering for eLS, one of these challenges was the lack of traditional requirements elicitation technique to get all of the stakeholders' requirements. They mentioned that the requirements engineers only focus on the technical requirements but not the social requirements of users. The paper suggested the need for an adequate requirements elicitation mechanism to detect and enhance users' social requirements to keep the users' sustainability of the eLS.

Tran and Anvari [22] highlighted the lack of the availability of a framework to address eliciting requirements of eLS' stakeholders, and confirmed the special nature of eLS because of the great number and diversity of stakeholders. The paper implied the need to open new insights in the perspectives of requirements elicitation by the software engineers, as well as the need for collaboration and communication in the requirements elicitation process. It focused on the questionnaire technique for corporate eLS in the context of Accounting Information Systems (AIS); a five-dimensional framework is proposed to guide the design of questionnaires that will be used in the requirements elicitation activity for eLS.

Ali and Lai [23] addressed the importance of communication and collaboration between stakeholders in the Global Software Development (GSD) context. We found that stakeholders in this context are comparable to stakeholders in eLS context, they are diverse in cultures, geographically dispersed, and there are times zones and language barriers, which made difficulties in engaging into an effective communication. Accordingly, there is a need of requirements elicitation approach to fulfill the collaborative needs and diverse context of stakeholders. The publication had presented a new method for requirements elicitation and analysis based on four stages involving some of the traditional requirements elicitation techniques e.g. use case, scenarios. Finally a preliminary evaluation was conducted through applying a case study on graduate students.

We can conclude that the surveyed papers confirms on the characteristics of the eLS and the need for new requirements elicitation approaches to overcome the limitations of the traditional ones in eLS context.

B. Crowdsourcing-Based Requirements Engineering

This section reviews and summarizes the use of crowdsourcing in requirements engineering, and presents the different crowdsourcing-based tools and approaches published in the requirements engineering literature. According to Hosseini [24], the four pillars of crowdsourcing are as follows, 1- The crowd: the people involved in a crowdsourcing action. 2- The crowdsourcer: the entity which looks for the power of the crowd for doing a task. 3- The crowdsourcing task: the activity or action in which the crowd participates. Finally, 4- The crowdsourcing platform: the system which a crowdsourcing task is accomplished within. Some platforms handle all of the software engineering phases; others handle specific phases. What is common between all platforms is that they all let a crowdsourcer find talents and benefit from reduced costs, solution diversity, creativity and problem solving [25]. There are many commercial platforms that are used in different research case studies or real-world projects in the field of software engineering, e.g. Upwork, TopCoder, Elance, Odesk, Utest, Amazon Mechanical Turk (AMT), IdeaScale, Stake Overflow [14].

Crowd-based requirements engineering as a term was coined by Groen et al. [4] to be a requirements engineering approach for acquiring and analyzing any kind of users' feedback from the crowd, with the aim of seeking validated user requirements. It was further elaborated by Groen and Koch [5] be "the combined set of techniques for analyzing data from the crowd using text and usage mining, motivational techniques for stimulating further generation of data, and crowdsourcing to validate requirements". Groen et al. [26] discussed the concept of crowd-based requirements engineering (CrowdRE) and its landscape and challenges to emphasize the use of it. The research mentioned the possible early return on investment from the use of crowdsourcing in requirements engineering; however there is a need for more empirical researches and case studies.

Crowdsourcing is used in requirements elicitation to help requirements engineering [27]. It assists in finding, detecting and involvement of different stakeholders who can outline software requirements. Crowdsourcing increases and improves the range of elicited requirements and, as a result, helps getting a whole idea of users' and other stakeholders' expectations from a software. Hosseini et al. [6] reported initial results in the use of crowdsourcing in requirements engineering after conducting two focus groups of experts to discover the crowdsourcing features and its quality attributes. The results showed that largeness, diversity, anonymous participation, volunteering and incentives lead to more correctness, and completeness of requirements gathering.

The use of crowdsourcing in requirements engineering is usually accompanied by a supported tool [4]. Groen et al. [4] presented a classification of the tools that were used to create a crowd-based requirements engineering approach: Social oriented collaboration tools, such as CrowdREquire. It was a platform that helped individuals and companies to find the best requirements specification for their proposed tasks and projects. It had a communication tool to connect requirements engineering professionals with the companies that request their services [28]. Srivastava and Sharma [12] proposed a

crowdsourcing-based tool to a case study on MyERP software to extract software requirements across various and different geographical crowd. The tool succeeded in getting diverse opinions, requirements and lowering the cost of it. Snijders et al. proposed and evaluated a requirement engineering method based on gamification and crowdsourcing called Crowd-Centric requirements engineering [11]. They also proposed Refine, which was a game-based online platform for requirements elicitation and refinement; it allowed the involvement of a crowd of stakeholders [29]. Sharma and Sureka [7] proposed CRUISE which was a platform that had a comprehensive vision to be used as a crowd-based platform for all requirements engineering activities. The platform was validated through a preliminary experimental study to investigate the feasibility and capability of the platform. Lim et al. [30] had proposed a novel method called StakeRare that uses social networks analysis to identify and prioritize requirements in large software projects. The method was based on building a social network of stakeholders and their recommendations of other stakeholders to reach a list of requirements using applied SNA measures. The system was evaluated by applying it on large size software project. The case study applied confirmed that StakeRare predicted stakeholder needs accurately and correctly prioritized.

Web-based approaches with social network analysis or recommender systems: for example StakeSource. It was a Web-based tool that automated stakeholder analysis. It crowdsourced the stakeholders themselves for recommendations about other stakeholders and aggregated their answers using social network analysis [31].

Text mining tools: focus on analyzing available data without actively involving stakeholders. Thereby not supporting elicitation directly, rather, they determined the relevance and importance of a sentence or statement through natural language algorithms, usually based on app store reviews [32], [33], [34]. Hosseini et al. [35] surveyed experts to support the use of crowdsourcing for the help of requirements elicitation. In [35] CRAFT proposed, it is a technique that utilized the crowd power to enrich text mining by allowing the crowd to categorize and annotate feedback through a context menu. This, in turn, helped in better identifying user requirements within forums feedback. Guzman and Maalej [33] studied the use of NLP and sentiment analysis in the field of requirements engineering and specifically for the elicitation activity. They analyzed different users' reviews in the mobile applications industry; they used a collection of analysis methods to reach for fine grained extracted features. Finally, their approach was evaluated by comparing the results with manually extracted features they obtained a *precision* up to 91% with 59% in average. NLP had been tested for its efficiency in comparison to the manual analysis of users' reviews by Groen et al. [36] the results showed that automated NLP analysis were much faster. Table 1 Lists the crowd-based tools in requirements engineering found in literature; it gave the tool name, its corresponding requirements engineering phase, the supportive tools and the reference citation.

TABLE. I. LIST OF LIST OF THE CROWD-BASED TOOLS IN REQUIREMENTS ENGINEERING

Tool Name	Requirements Engineering Phase	Supportive Tool	Ref. No
CrowdREquire	Requirements Specification	-	[28]
-----	Requirements Elicitation	SNA	[12]
Refine	Requirements Elicitation	Gamification	[29]
CRUISE	Requirements Engineering Phases	-	[7]
StakeRare	Requirements Elicitation & Prioritization	SNA	[30]
-----	Requirements Evolution	NLP	[32], [33]
CRAFT	Requirements Elicitation	Expert Survey	[35]
StakeSource	Stakeholder Analysis	SNA	[31]
Requirements Bazaar	Requirements Elicitation & Prioritization	Social SW	[37]
CrowdCentricRE	Requirements Elicitation	Gamification	[11]

C. The use of Crowdsourcing in eLS

This section presents the current use of crowdsourcing in eLearning as available in the literature. In the eLS literature the use of crowdsourcing is concerned with the crowdsourcing of the creation of the learning content [38] and providing the learning service to a large number of users, e.g. Coursera that has classes with thousands of students [39], [40]. Although we are concerned with the use of crowdsourcing for requirements elicitation for eLS, we will present in this section how crowdsourcing were used in general in the eLS literature for completeness of coverage. Suhonjic et al. [41] proposed a crowdsourcing model that combines the collaborative learning and crowdsourcing mechanisms to implement it on learner-centered approach. The study aimed to enhance the participation and collaboration of learners as learning creators. The study was evaluated by case study applied on 74 students on Belgrade University. It showed an enhancement in user participation, and good quality of learning contents. Tarasowa et al. [42] presented Crowd-Learn which was the use of learning objects for structured eLS to support the system and to manage the learning objects (files, presentations). The model used the wiki style collaborative authoring, and crowdsourcing for the creation of learning contents of eLS. The system evaluated by case study applied on an information system lecture at Chemnitz Technical University. The Wiki slides are structured within the lecture and added questions for student self-assessment before the final exam. Tarasowa found that eLearning material when combined with crowd-sourcing and collaborative social approaches can help to cultivate innovation by collecting and expressing different individual's ideas.

Barbosa et al. [43] studied crowdsourcing for Massive Online Open Course (MOOC) and set dimensions to classify the types of crowdsourcing for eLearning tools, and identified many crowdsourcing tools for eLearning. The publication discussed and classified 22 crowdsourcing eLearning tools found in the Internet e.g. Coursera, Udacity and MIT

OpenCourseW. The tools were varying from online universities to marketplaces for online courses. It built a framework to compare and group the tools, using a set of eleven dimensions. Collaboration is a key aspect, as in a class of thousands of students, it was virtually impossible for a teacher to give attention to every single student. The crowd must help itself to enable this approach, moreover, it offers learning in a natural way. The job of the teacher becomes more illustrative, and less evaluative.

Paulin & Haythornthwaite [40] addressed how the evolution of the Web changes how, where and with whom people learn, and the opportunities and challenges this rises for the future of educational practice. The publication focused on taking advantage of crowdsourcing to create and manage large-scale learning enterprises. MOOCs were the principal point for large-scale online learning. The power of the crowd was being leveraged to address many of the scale-related issues that arise in MOOCs. Elements suggested to be addressed by crowdsourcing were content, discussion, evaluation, behavior, practices, learning analytics, and assessment and feedback.

Karataev and Zadorozhny [38] studied the crowdsourcing of learning content to anyone. They introduced a novel framework for social learning that allows any person to author educational content as mini-lessons, learn lessons by use adaptive learning pathways to learn lessons, and interact with their peers. The system was evaluated through a number of classroom studies. The results showed that adaptive social learning can be utilized by collective learning experiences also they found that students with very high similarity tend to arrange groups.

We can conclude that literature survey ensures that there is a gap in the requirements elicitation activity for the eLS, there is a need for a new or enhanced approach to fill in this gap. Also, the literature survey gives a motivation and recommendation to use crowdsourcing in requirements engineering as an emerging approach, hence we present Crowdsourcing based Requirements Elicitation for eLS (CREeLS). The following section presents detailed overview of our proposed framework and approach CREeLS.

III. CREELS

This section presents CREeLS; our proposed automated approach for Crowdsourcing based Requirements Elicitation for eLS. It is made up of a framework, which presents the broad lines and the basic concepts, and an automated approach.

A. CREeLS' Framework

After reviewing the literature we came up with the recommended sources of crowd in the eLS context, and the supported tools that can utilize the crowd interactions or opinions needed in the requirements elicitation activity. We could form the proposed framework based on these sources and tools. The crowd sources are interactivity in the use of the power of social networking and applying text mining tool for the received feedback. The framework gives broad lines for achieving crowd-based requirements elicitation for eLS. The use of each suggested tool in crowdsourcing for requirements engineering is separately evaluated in the literature. The proposed framework is illustrated in Fig. 1.

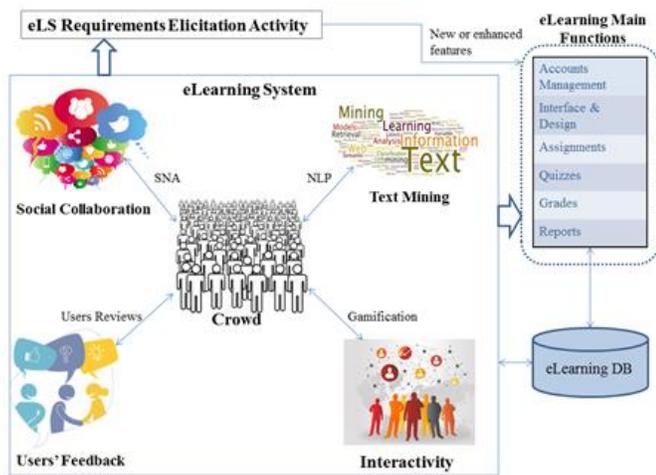


Fig. 1. The Proposed Framework of the Necessary Elements of CREeLS.

1) *eLearning system*: The proposed framework is operating in the context of eLS. The suggested elements of crowdsourcing requirements elicitation for eLS are established inside the eLS, as they are parts of the eLearning process as we will explain in the next sub-sections. Those elements are considered inputs for the eLS requirements elicitation activity, the results from the analysis of those elements can reveal new or enhanced features for the eLS functions.

2) *The crowd*: The crowd in crowdsourcing context is the group of people who engage and participate in the crowdsourcing activity. According to Hosseini et al. [24] crowd is characterized by: diversity, suitability, anonymity, largeness, and undefined-ness. Crowd in eLS are the eLS users from learners, instructors, administrators, management, or parents or learners. The framework can be applied on the context of crowdsourcing platforms. Crowdsourcing platform connects requesters with online workers [25].

3) *Users' feedback*: The target of the proposed framework is to minimize the gap between the development team and the eLS' stakeholders. Feedback is one source of getting what's in the users' minds without intentionally getting in direct interaction with them [44]. There must be different feedback methods from the interaction between the eLearning participants in the different forms of social collaborations that support eLS. Learning is a lifelong process that requires continuous feedback and adaptation. Feedbacks can be on the eLS itself, course and its material, or the instructor and management.

Users' comments or reviews are means of feedback that can be found in the eLS in its different modules. Users' comments and reviews can contain useful information for developers; they include good, bad, or recommended features [45]. Thus, the analysis of these reviews is important for the requirements engineering activities [46]. The proposed framework suggests the use of feedback analysis methods to gain the benefits from eLS users' comments and reviews in requirements elicitation activity while developing eLS.

4) *Interactivity*: Interactivity between the eLearning participants (Instructor, learner, course, and management) is important. Interactivity leads to better course results for learners [47], and to know the participants' opinions in the different modules of the eLS, e.g. course material, quizzes, assignments, scheduling, eLearning process, and participation between the different eLS users. Improving social interaction in eLS can improve user satisfaction. Social interactions involve more collaborative activities. Crowdsourcing in eLS not only increases the amount of educational content but also improves its quality [42].

The collaboration of a person's contribution with a larger, shared, cooperative work is a type of interaction, thus the act of crowdsourcing can be considered naturally interactive. Jennifer and Brigid [48] identified three possible categories of interactive, crowdsourced works: Category 1: Linear/single-channel works created from multiple user contributions. Category 2: Interactive works created from unique individual contributions. Category 3: Interactive works created from multiple user contributions.

Category 3 is recommended to be the first to start with; as many stakeholders are involved in the requirements elicitation process. Under this category we suggest the use of gamification in the interactivity element of the framework. Gamification seeks of integrating the game process and techniques in a non-gaming process to be more attractive. Gamification seeks out for improvement of the user's performance, commitment, and motivation [49].

5) *Text mining tools*: Text mining is the process of analyzing unstructured text using data mining techniques. In the framework we use text mining in order to analyze eLearning participants' written interactions, which exist in the different modules in the eLS. It can be found in social networking applications attached to the eLS, discussion forums, comments spaces below blogs posts, or any other different posts. The use of text mining techniques will extract the hidden requirements. One method of text mining analysis is Natural Language Processing (NLP). Text mining can be used in requirements elicitation process for eLS to identify new, good, bad or need to be enhanced features [33].

6) *Social collaboration*: Social Network Sites (SNS) are an example of social software; they are used for communicating and connecting with others—anytime and anywhere. SNS allow the creation of social groups, where many people with similar interests are connected together and communicate in different forms. Studying the patterns of social collaborations in SNS is a method of getting the users requirements and knowing their behavior.

Social Network Analysis (SNA) is a method of understanding the relationships among the nodes of interactions and studies the patterns and effects of the relationships. Thus, SNA can help in requirements elicitation to identify eLS requirements.

B. CREeLS Automated Approach

The authors are encouraged to propose new requirements elicitation approach for eLS based on the crowdsourcing concept because of the following factors; eLS characteristics, the crowd-based requirements engineering definitions mentioned earlier, the findings from the literature review. In addition to, the proposed framework discussed in the previous section, the need of an automated approaches that combines both the social and technical aspects of software engineering in general and requirements elicitation in particular [50].

CREeLS approach has five phases, (1) Creating a channel for users to post their feedback, or show their interactions. (2) Extracting users' interactions or feedback. (3) Analyze users' interactions or feedback. (4) Evolve fine software requirements. (5) Categorize and consolidate the requirements. eLS stakeholders are CREeLS's crowd.

Phase 1: Creating (a) channel(s) for users to post their feedback or show their interactions. The approach starts by creating a channel(s) or a facility (ies) for users to post their feedback. The requirements engineer decides to select one, or more than one channel to be created. The recommended available channels are: a) allow users to post their feedback through posting reviews on the use of their eLS. This can take place through surveys, or evaluation buttons in the eLS applications. The paper will focus on this method. b) allow the use of social networking sites (SNS) within the eLS, or add social networks application to the eLS. c) create a gamified way to encourage users to post their experience of using the eLS. The use of gamification technique is promising in the requirements engineering field [11]. The use of incentives, collected points, and badges are different techniques of gamification. There is also one more hidden channel, which is creating different interactivity tools for the eLS users. The analysis of these interactions, eLS requirements can be extracted.

The previous channels are crowdsourcing the requirements elicitation task intentionally and unintentionally. Channel number one is considered as a direct channel where users know and intend to post their feedback. Channel two is considered an indirect way of extracting feedback as in the use of SNS; users post their feedback unintentionally through threads of discussions for socializing and communicating among different members or administrators. Channel three can be considered as both a direct and indirect way. It can be direct because users may know about the game purpose for collecting reviews, and intend to participate in it. On the other hand, for the users who like to be involved in games, they participate in it for the love of games, and unintentionally post their reviews. This phase allows eLS stakeholders to freely disclose their requirements on their way, no matter the number of feedback posts, geographical location, culture or background.

Phase 2: Extract users' interactions or feedback. Users' feedback posted in phase one should be collected and extracted in phase two. Users' feedbacks are stored in the eLS databases so the database administrator is involved in this phase to collect the stored feedback. The requirements engineer decides the data needed and ask the database administrator to prepare it and then send it to the text analyst to perform the analysis techniques.

Phase 3: Analyze Users' interactions or feedback. Analysis of users' feedback can be accomplished using different ways; text mining analysis is an example and it is under experimentation in this paper. The use of natural language processing in analyzing users' feedback; applying text preprocessing, then the feature extraction algorithms to extract users requirements.

Phase 4: Evolve fine software requirements. An optimization functions for natural language processing algorithms are applied to extract fine software requirements. Optimization functions intend to find the best values that achieve the highest coherence value and best collection of keywords and topics that represent the eLS requirements.

Phase 5: Categorize and consolidate the requirements. This phase is accomplished manually by the requirement engineer. Fine software requirements output from phase four are considered as input for this phase, requirements engineer classifies and categorizes software requirements then merge similar requirements together to have the final form of the eLS users' requirements.

IV. EXPERIMENTAL STUDY

This section presents our experimental study for CREeLS. According to Wohlin et al. [51], an experiment gives more control over the situation; it allows to compare the results when one variable is changing and the others are fixed. We wanted to analyze eLS users' feedback and evaluate whether this feedback was truly representing eLS requirements, then we will check whether the extracted requirements were similar and coherent as the manually extracted users' requirements. The experimental study reflects phase 3 in CREeLS approach, as we wanted to test whether CREeLS will succeed to extract keywords that reflect LMS users' requirements in less time than the manual extraction of requirements. Phases 1, 2, 4 and 5 will be skipped because of the following. Phase 1 requires programming development into the LMS to attach one of the suggested facilities tools, and this is out of the current research scope as we are focusing on LMS requirements extraction from users' reviews. Phase 2 is responsible for collecting and extracting the required reviews from the LMS database; we substituted this phase with the dataset we acquired from G2Crowd. Phases 4 and 5 will be left for future publication when we enhance CREeLS.

The experimental study started by data collection and went through the requirements extractions steps as detailed below. The only data we could be able to collect is LMS users' reviews from G2Crowd company; the data is on an Excel sheet. The reviews are for more than 20 products for both LMS for education purpose, e.g. universities and schools, and LMS for corporate purpose. The reviews are classified into reviews on features that users like, and reviews on features that users dislike. We used only one product for the current study which is 'Blackboard' with 5036 users' reviews. We followed the following steps for eLS feature extraction. The steps are illustrated in Fig 2. We should mention that it is an evolutionary approach in which the results are at a certain point in the requirements elicitation activity, because there is a continuous stream of information from an LMS.

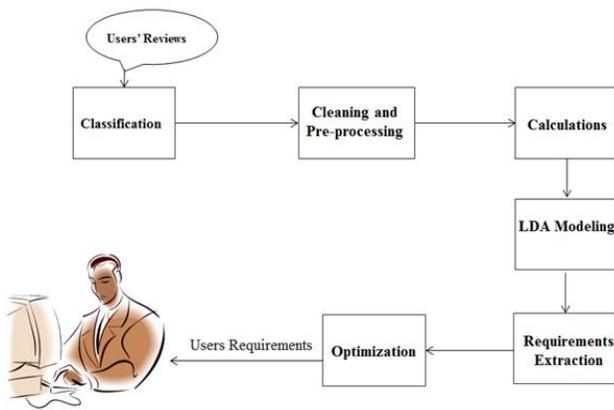


Fig. 2. Requirements Extraction Steps.

1) *Classification*. The dataset was classified according to the LMS type; features which users like, features that users dislike, and the role of the respondent, e.g. administrator, student, or instructor. In our study we selected first LMS products for educational purpose, because they have a larger number of reviews than corporate LMS. Second, we selected reviews about features which users dislike, because this will imply the users' needs and their requirements for the LMS. Third, role of users, this criterion we couldn't consider as most of the users were students and only few reviews were for administrators or instructor or management. Because text analysis needs a large amount of text for better results, we decided to include all the reviews with no classification according to users' roles.

2) *Cleaning and Pre-processing*. To perform the feature extraction, raw data should be prepared so that it can be easily analyzed and to get better results. Preparation of data should be performed in terms of: 1- cleaning or removing the special characters from the text e.g. commas, dashes, or semi colons, because these special characters were not needed in the extraction process. 2- Tokenization which is splitting the text into words, so that we can treat each word separately. The processes of cleaning and tokenizing text were performed together in one step using one function. 3- Stopwords removal: in computing, stop words are common words that has little value in the text [52] e.g. and, is, are. We used the standard list of stopwords provided by Gensim library and we added words that we found common in users' reviews, but weren't describing features e.g. "like", "dislike", "there", "easy". 4- Part of Speech Tagging (POS): it is tagging each word in the text as its corresponding grammatical part of speech [53]. We used the POS of wordnet in NLTK Package, in this step we defined and extracted the nouns, verbs, adjectives, and adverbs in the text. Because, it was important to understand what a text is about. 5- Stemming refers to a process that removes the ends of words to reduce it to its base or root form. 6- Lemmatization: aims to remove inflectional endings (are letters at the end of a word that change its meaning) and to return the word to its base or dictionary form.

3) *Words calculations*: We created a dictionary for each word used in the text. The output from this function was the minimum number each word has appeared in the text, and the maximum ratio of each word appeared in the text. We used the Gensim dictionary function in this step. TFIDF weight measurement was used for extracting features from the users' reviews. TF-IDF stands for Term Frequency-Inverse Document Frequency, the tf-idf weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus. The importance increased proportionally to the number of times a word appeared in the document but was offset by the frequency of the word in the corpus. Variations of the tf-idf weighting scheme are often used by search engines as a central tool in scoring and ranking a document's relevance given a user query [54]. Typically, the tf-idf weight is composed by two terms. The first term computes the normalized Term Frequency (TF); the number of times a word appears in a document, divided by the total number of words in that document. The second term is the Inverse Document Frequency (IDF), computed as the logarithm of the number of the documents in the corpus divided by the number of documents where the specific term appears.

4) *Latent Dirichlet Allocation (LDA) modeling*: LDA is a topic modeling algorithm; it is a generative probabilistic algorithm for data collection, in our case is LMS users' reviews. In the context of topic modeling, each topic is considered as a group of topics [55]. This means that each user review can have more than one requirement (topic) associated; also, each requirement (topic) can have more than one keyword associated to it. In this step we used LDA algorithm to extract the top keywords in the text based on the calculations done in the step 3, which was calculating the weight of each word in each review and in the whole text. The result of applying LDA was the top feature keywords in the whole text.

5) *Requirements extraction*: Based on the top keywords in the text, we applied function that categorized these keywords into number of topics; these topics represented the users' requirements, and composed of number of the top keywords and its percentage of relevance to this requirement.

6) *Optimization*: We performed an optimization function, which specified the optimum number of requirements in the text. This function calculated the coherence value of the top keywords in text that composed a requirement and the distance between the requirements to reduce the overlapping between the requirements keywords. It also calculated the optimum value for keywords in terms of the minimum number each word appeared in the text, and the maximum ratio of this word appeared in the text. The results from the optimization function were 8 topics with keywords with minimum repetition in the text = 5 times, and maximum ratio to appear in the text is 40% with coherence value of 0.47. Each topic has dominant 10 keywords which indicated the system requirements. The topics distribution is illustrated in Fig. 3.

We noticed that the topics circles were not overlapped which means that the topics keywords were not overlapped, each topic represented its own keywords and topic. Circles with numbers represented the topics, the size of the circle indicated the dominance of this topic within the text, the keywords of the topic appeared in the text, and the highest frequency keyword appeared at the first. Table 2 illustrates the dominant keywords in each topic. We made some of these keywords bold to emphasize them, as we believe that they have higher tendency on indicating some of the eLS users' requirements.

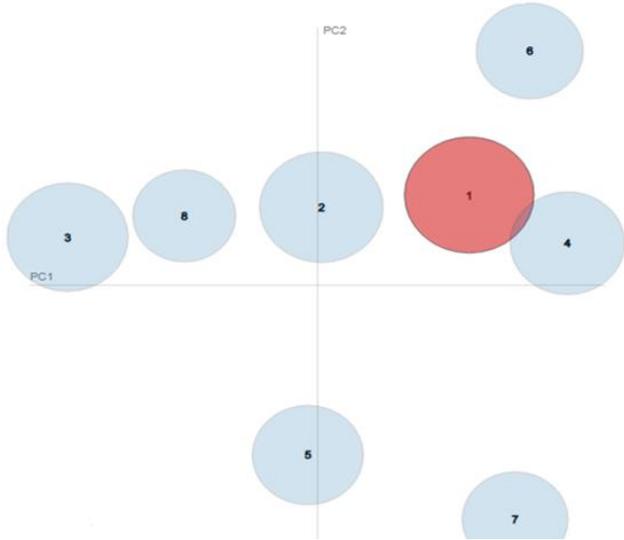


Fig. 3. Extracted Number of Topics.

TABLE. II. TOPICS AND KEYWORDS

Topic No.	Keywords
1	issue, content, time , navigation
2	mobile , time, confuse, phone
3	Interface , grade, clunky, look
4	design , layout, program, difficult
5	assignment , able, think, class
6	time , maintenance, fact, hour
7	know, difficult , product, specific
8	buggy , group, calendar, glitch

V. EXPERIMENTAL STUDY EVALUATION

In this section, the evaluation process of the experimental study is presented. Then a discussion of the results, its interpretations, limitations and threats to validity are discussed.

A. Results

There is no better way to accurately evaluate the topics generated from our approach, rather than manually, using human brain to revise the available users' reviews. We manually evaluated the approach by analyzing and reviewing each user's review for each LMS product to extract the user requirements for each review. Then we counted the number of reviews relevant to each requirement, and then we got the percentage of frequency for each requirement by dividing the number of reviews related to one requirement by the total number of reviews. This calculation was repeated for each

manually extracted user requirement. The topics were sorted by importance from the highest percentage to lowest percentage. Finally, we compared it with the extracted topics. The process of manually extracting the users' requirements from the LMS reviews was accomplished by one person familiar with requirements engineering and extracting user requirements.

We evaluated the results of our automatic topic modeling by using *precision*, *recall*, and *F-measure*. According to Brownlee [56], *Precision* is how many selected items are relevant; it is computed by dividing the number of true positives by the sum of true positives and false positives. *Recall* is how many relevant items are selected; it is computed by dividing the number of true positives by the sum of true positives and false negatives. *F-measure* tests the experiment's accuracy; it's approximately the average of both *recall* and *precision*, the best value is 1 and the worst is 0. *F-measure* is calculated as follows:

$$F - measure = 2 * \frac{Precision * Recall}{Precision + Recall}$$

True positive is the requirement that is both manually and automatically identified, false positive is the requirement that is automatically but not manually identified, and finally false negative is the requirement that is manually but not automatically identified from the approach. While working on the experimental study we considered a feature as true positive, if it was automatically extracted from a review and was also manually identified in that review. False positives are features that were automatically associated to a review in one of the topics, but were not identified manually in that review. Finally, false negative features were manually identified in a review but were not present in any of the extracted topics associated to the review. The results were as follows; *precision* 0.66, *recall* 0.77 and *F-measure* 0.71. In addition to *precision*, and *recall*, there is the *coherence* of topics which assesses how well the topics are logical and consistent and whether they share a common theme; its value reached up to 0.47. Those results were at certain points when running the approach; any changes in text reviews, or their number could affect the final results.

B. Results Interpretations

The quantitative results and the qualitative evaluation of the proposed approach were positive. CREeLS succeeded to extract an adequate number and good representation of topics, which are users' requirements of the LMS in a small time frame compared to the manual process. The qualitative evaluation showed coherent topics; most of the keywords represent LMS features, but not noise, and the topics' keywords were relevant to the LMS requirements. The results of the proposed approach can be manually adjusted for better users' requirements understanding. Synonym keywords and duplicate topics represented the importance of this topic as an LMS product requirement, e.g. *interface*, *design*, *look*, *clunky*, and *layout*, reflect the need for better and modern design for the LMS. The keyword 'time' appeared in three topics but in different contexts; in topic 1 with the keyword navigation, it means that the navigation process takes time from the user. Also, in topic 2, the keyword *time* appeared with *mobile*, *phone*, and *confuse* means that the LMS version on the mobile takes more time. Finally, in topic 6 the keyword 'time'

appeared with *maintenance* and *hour*, which means that the *maintenance time*, is not adequate with users. All of these interpretations for the different contexts of using the word *time* were supported by the manual evaluation of the results.

C. Limitations

This section gives some limitations of CREeLS. First, the features that are non-frequently mentioned in the reviews text are most probably not detected, as there is a minimum count for the keywords to appear in the reviews text. Second, the resulted topics keywords are not correlated. Third, LDA needs large number of reviews to perform better. These limitations could be improved by encouraging users to write their reviews, adding language patterns and the use of two connected keywords and not a single one only. This can allow the identification of low occurring keywords and add more correlation to the topics' keywords, and taking context into consideration.

D. Threats to Validity

The process of manually extracting the users' requirements from the LMS reviews was accomplished by one of the authors, and not an actual business analyst or requirements engineer. The author can have misunderstanding or incomplete information about real eLS. Also biased implications on the manual evaluation results can be present.

VI. CONCLUSION AND FUTURE WORK

This paper presented CREeLS, the crowdsourcing based requirements elicitation approach for eLS. CREeLS is made up of a framework for the necessary elements of crowdsourcing suggesting specific tools for each element, and a phased approach to implement the framework. The phased approach uses topic modeling techniques to extract requirements from eLS users' feedback. An experimental study was conducted to extract users' requirements from real-life LMS users' reviews. The results were evaluated by manually revising the user's reviews which showed coherence of topics up to 0.47, *recall* value 0.79, *precision* 0.66 and *f-measure* of 0.71. The results were very promising; therefore, we contend that CREeLS can help requirements engineers for eLS to analyze users' opinions and identify the most common users' requirements for better software evolution. For our future work we intend to continue improvement and validation of CREeLS through trying other topic modeling techniques that can combine two or three words to give more understandable results of the users' requirements. In addition, further experimental studies will be carried out on a greater number of text reviews for more than one LMS product, and will also include the corporate LMS products.

ACKNOWLEDGMENT

We are very grateful to G2Crowd Company for sharing with us its LMS' users' reviews data.

REFERENCES

- [1] [(2019, May) Cambridge English Dictionary. [Online]. <https://dictionary.cambridge.org/dictionary/english/crowd>.
- [2] J.Howe. (2006, June) Wired "The Rise of Crowdsourcing". [Online]. http://archive.wired.com/wired/archive/14.06/crowds.html?pg=4&topic=crowds&topic_set=.
- [3] J. Howe. (2006, June) Crowdsourcing: A Definition. http://crowdsourcing.typepad.com/cs/2006/06/crowdsourcing_a.html.
- [4] E. C. Groen, J. Doerr, and S. Adam, "Towards Crowd-Based Requirements Engineering A Research Preview," in Requirements Engineering: Foundation for Software Quality, Samuel A. Fricker and Kurt Schneider, Eds.: Springer International Publishing, 2015, pp. 247-253.
- [5] E. Groen and M. Koch. (2016, June) How Requirements Engineering can Benefit from Crowd. re-magazine.ireb.org "Requirements Engineering Magazine".
- [6] M. Hosseini, K. Phalp, J. Taylor, and R. Ali, "Toward Crowdsourcing for Requirements Elicitation: Results from Expert Survey," in Proceeding of The 20th International Working Conference on Requirements Engineering: Foundation for Software Quality, Essen, Germany, April, 2014.
- [7] R. Sharma and A. Sureka. (2017) CRUISE: A Platform for Crowdsourcing Requirements Elicitation and Evolution. [Online]. <http://www.robo-paathshaala.in/ashish/IC3-2017-CRUISE-PAPER.pdf>.
- [8] K. Mao, L. Capra, M. Harman, and Y. Jia, "A Survey of the Use of Crowdsourcing in Software Engineering," Journal of Systems and Software, vol. 126, pp. 57-84, 2016.
- [9] I. Sommerville, Software Engineering, 9th ed. USA: Pearson, 2011.
- [10] D. Zowghi and C. Coulin, "Requirements Elicitation: A Survey of Techniques, Approaches, and Tools," in Engineering and Managing Software Requirements. Berlin: Springer, 2005, pp. 19-46.
- [11] The Standish Group, "Chaos Report," The Standish group, 2018.
- [12] R. Snijders, A. Ozum, S. Brinkkemper, and F. Dalpiaz, "Crowd-Centric Requirements Engineering: A Method Based On Crowdsourcing And Gamification," Utrecht University, Utrecht, The Netherlands, Technical Report ISSN: 0924-3275, 2015.
- [13] P. K. K. Srivastava and R. Sharma, "Crowdsourcing to Elicit Requirements for MyERP Application," in Proceeding of 1st International Workshop on Crowd-Based Requirements Engineering (CrowdRE), Ottawa, 2015, pp. 31-35.
- [14] N. Abdul Rahman and S. Sahibuddin, "Challenges in E-learning: an Requirements Engineering Perspective," in Proceedings of the 3rd International Conference on Computing and Informatics, Bandung, Indonesia, 2011.
- [15] Australian Flexible Learning framework, "Practical Guide to E-learning for Industry," Australian Flexible Learning framework, 2007.
- [16] B. Holmes and J. Gardner, E-learning: Concepts and Practice.: Sage Publications, 2006.
- [17] N. M. Rizk, E. S. Nasr, and M. H. Gheith, "Requirements' elicitation needs for eLearning Systems," in On Proceeding of The 12th International Computer Engineering Conference (ICENCO), Giza, Egypt, 2016.
- [18] W. Horton and K. Horton, E-learning tools and techniques.: Wiley, 2003.
- [19] G.I Gheorghiu. (2017, November) The Content in a Learning Content Management System (LCMS). SelectHub.
- [20] S. Irlbeck and J. Mowat. (2007, January) Learning Content Management System (LCMS). http://www.herridgegroup.com/pdfs/LCMS_chptr_Aug29%202005_%20Final%20Sept%206%2005.pdf.
- [21] S. B. AlKhuder and F. H. AlAli, "E-Learning Systems Requirements Elicitation: Perspectives and Considerations," International Journal of Virtual and Personal Learning Environments (IJVPLE), vol. 7, no. 1, pp. 44-55, January-June 2017.
- [22] H. Minh Thi Tran and F. Anvari, "A Five-Dimensional Requirements Elicitation Framework for e-Learning Systems," International Journal of Information and Electronics Engineering, vol. 6, no. 3, pp. 185-191, 2016.
- [23] N. Ali and R. Lai, "A method of requirements elicitation and analysis for Global Software Development," Journal of Software Evolution and Process, vol. 29, no. 4, pp. 1-27, 2017.
- [24] M. Hosseini, K. Phalp, J. Taylor, and R. Ali, "The Four Pillars of Crowdsourcing: a Reference Model," in Proceedings of IEEE Eighth International Conference on Research Challenges in Information Science (RCIS), Marrakech, 2014, pp. 1-12.

- [25] A. L. Zanatta, L. S. Machado, G. B. Pereira, R. Prikładnicki, and E. Carmel, "Software Crowdsourcing Platforms," *IEEE Software*, vol. 33, no. 6, pp. 112-116, November 2016.
- [26] E. C. Groen et al., "The Crowd in Requirements Engineering: The Landscape and Challenges," *IEEE Software*, vol. 34, no. 2, pp. 44-52, March 2017.
- [27] M. Hosseini, A. Shahri, and F. Dalpiaz, "Configuring Crowdsourcing for Requirements Elicitation," in *In Proceeding of The IEEE Ninth International Conference on Research Challenges in Information Science*, Greece, 2015.
- [28] A. Adepetu, K. A. Ahmed, and Y. Al Ab, "CrowdREquire: A Requirements Engineering Crowdsourcing Platform," Association for the Advancement of Artificial Intelligence, Technical Report 2012.
- [29] R. Snijders et al., "REfine: A Gamified Platform for Participatory Requirements Engineering," in *IEEE 1st International Workshop on Crowd-Based Requirements Engineering (CrowdRE)*, Ottawa, Canada, 2015.
- [30] S. L. Lim and A.y Finkelstein, "StakeRare: Using Social Networks and Collaborative Filtering for Large-Scale Requirements Elicitation," *IEEE Transactions on Software Engineering*, vol. 38, no. 3, pp. 707-735, 2012.
- [31] S. L. Lim, D. Quercia, and A. Finkelstein, "StakeSource: Harnessing the Power of Crowdsourcing and Social Networks in Stakeholder Analysis," in *In Proceedings of the IEEE 32nd International Conference on Software Engineering*, 2010.
- [32] T. D. D. Breaux and F. Schaub, "Scaling Requirements Extraction to the Crowd: Experiments with Privacy Policies," in *On Proceeding of IEEE 22nd International Requirements Engineering Conference (RE)*, Karlskrona, Sweden, 2014, pp. 163-172.
- [33] E. Guzman and W. Maalej, "How Do Users Like This Feature? A Fine Grained Sentiment Analysis of App Reviews," in *Proceeding of Requirements Engineering Conference (RE)*, 2014 IEEE 22nd International, Karlskrona, Sweden, 2014, pp. 153-162.
- [34] C. Wang, M. Daneva, M. van Sinderen, and P. Liang, "A systematic mapping study on crowdsourced requirements engineering using user feedback," *Journal of Software: Evolution & Process*, pp. 1-20, July 2019.
- [35] M. Hosseini, E. C. Groen, A. Shahri, and R. Ali, "CRAFT: A Crowd-Annotated Feedback Technique," in *IEEE 25th International Requirements Engineering Conference Workshops*, Lisbon, Portugal, 2017.
- [36] E. C. Groen, J. Schowalter, S. Kopczynska, S. Polst, and S. Alvani, "Is there Really a Need for Using NLP to Elicit Requirements? A Benchmarking Study to Assess Scalability of Manual Analysis.," in *REFSQ 2018 Joint Proceedings of the Co-Located EventsAt: Utrecht, The Netherlands* Volume: CEUR 2075, Utrecht, Netherlands, 2018.
- [37] D. Renzel and R. Klamma, "Requirements Bazaar: Open-Source Large Scale Social Requirements Engineering in the Long Tail," *IEEE Computer Society Special Technical Community on Social Networking E-Letter*, vol. 2, no. 3, September 2014.
- [38] E. Karataev and V. Zadorozhny, "Adaptive Social Learning Based on Crowdsourcing," *IEEE Transactions on Learning Technologies*, vol. 10, no. 2, pp. 128 - 139, April-June 2017.
- [39] C.s E. Barbosa et al., "Conceptual Crowdsourcing Models for E-Learning," in *On Proceeding of IEEE International Conference on Systems, Man, and Cybernetics*, San Diego, 2014, pp. 1121-1127.
- [40] D. Paulina and C. Haythornthwaiteb, "Crowdsourcing the curriculum: Redefining e-learning practices through peer-generated approaches," *The Information Society*, vol. 32, no. 2, pp. 130-142, 2016.
- [41] A. Z. Suhonjić, M. Despotović-Zrakić, A. Labus, Z. Bogdanović, and D. Barać, "Fostering students' participation in creating," *Interactive Learning Environments*, vol. 27, no. 1, pp. 72-85, March 2019.
- [42] D. Tarasowa, A. Khalili, S'oren Auer, and J. Unbehauen, "CrowdLearn: Crowd-sourcing the Creation of Highly-structured E-Learning Content," *International Journal of Engineering Pedagogy*, vol. 5, no. 4, 2015.
- [43] C. E. Barbosa, V. J. Epelbaum, M. Antelio, J. Oliveira, and J. M. Souza, "Crowdsourcing Environments in e-Learning Scenario: A classification based on educational and collaboration criteria," in *On Proceeding of IEEE International Conference on Systems, Man, and Cybernetics*, 2013, pp. 687-692.
- [44] T.Wu. (2015, June) the Right State of Mind. Help Scout.
- [45] D. Pagano and W. Maalej, "User FeedBack in Appstore an Empirical Study," in *in Proceeding of Requirements Engineering Conference*, Rio de Janeiro, 2013, pp. 125- 134.
- [46] N. M. Rizk, A. Ebada, and E. S. Nasr, "Investigating Mobile Applications' Requirements Evolution Through Sentiment Analysis of Users' Reviews," in *Proceeding of 11th International Computer Engineering Conference (ICENCO)*, Cairo, 2015, pp. 123-130.
- [47] C. Palazuelos, D. García-Saiz, and M. ZorriI, "Social Network Analysis and Data Mining: An Application to the E-Learning Context," in *Computational Collective Intelligence. Technologies and Applications*, Costin Bădică, Ngoc Thanh Nguyen, and Marius Brezovan, Eds. Craivo, Romania: Springer Berlin Heidelberg, 2013, pp. 651-660.
- [48] J. Proctor and B.d Maher. (2013) Emotional Multiplicities in Multi-Sourced Work. <http://dnaanthology.com/anvc/dna/emotional-multiplicities-in-multi-sourced-work>.
- [49] O. Pedreira , F. García , and N. Brisabo, "Gamification in software engineering – A systematic mapping," *Information and Software Technology*, vol. 57, pp. 157-168, 2015.
- [50] G. Baxter and I. Sommerville, "Socio-technical systems: From design methods to systems engineering," *Interacting with Computers*, vol. 23, no. 1, pp. 4-17, January 2011.
- [51] C. Wohlin et al., *Experimentation in Software Engineering.*: Springer, 2012.
- [52] (2019, April) Stanford University. [Online]. <https://nlp.stanford.edu/IR-book/html/htmledition/dropping-common-terms-stop-words-1.html>
- [53] (2019) Sketch Engine. [Online]. <https://www.sketchengine.eu/pos-tags/>
- [54] TFIDF. [Online]. www.TFIDF.com.
- [55] D. M. Blei, A. Y. NG, and M. I. Jordan , "Latent Dirichlet Allocation," *Journal of Machine Learning Research* , vol. 3, pp. 993–1022, March 2003.
- [56] J. Brownlee. (2014, March) Classification Accuracy is Not Enough: More Performance Measures You Can Use. machinelearningmastery.com.