REQUIREMENTS ENGINEERING IN THE WORLD OF APPS

A prototype for discovering and visualizing requirements from user forums

By

AMASI S. H. ELBAKUSH

A thesis submitted in fulfillment of the requirements for the degree of

MASTER OF BUSINESS INFORMATICS

Under the supervision of

Dr. Prof. Fabiano Dalpiaz
Dr. Prof. Sjaak Brinkkemper
Mr. Johan Schlingmann

UTRECHT UNIVERSITY

Department of Natural Science

OCTOBER 2018

GRATITUDE

To my Family, my supporting husband and patient son, to my brilliant advisor who endured my frustrations and breakdowns and never stopped pushing me forward, to Stabiplan, Johan and Rene, for giving me this great knowledge journey.

To you all, a million thanks from the bottom of my heart.

Amasi Elbakush

October-2018, Nieuwegein

Table of Contents

ABSTRACT

We are living in a world that is rapidly growing digitally, and businesses are becoming increasingly dependent on information. In order to adapt to this growth, and gain competitive advantage, businesses seek new innovative approaches and communication channels to extract new software requirements from online user data. Previous and current research mainly focus their efforts on exploring social media and mobile application platforms for requirements discovery and extraction. In this research, we focus on exploring requirement extraction for desktop applications from technical user forums. We build a prototype tool for scraping a user forum, processing the textual data with NLP tools, and visualizing the resulting data using a visual analytics tool. The classification accuracy on the data set is between 60-90% while the recall between 80-90%. The Naïve Bayes Classifier outperformed other binary classifiers for the data of this research domain. We conducted experiments and interviewed experts to evaluate the perceived usefulness of our prototype. Results show positive feedback on our prototype as effective and efficient tool to support product managers discovering requirements and new markets.

Keywords: User feedback, Autodesk forum, requirement discovery, NLP, data mining, perceived usefulness

1. INTRODUCTION

A crucial part of software project development relies on successfully performing requirements engineering (RE) activities. One of these activities includes requirements elicitation. Eliciting requirements for software projects is a challenging and important task. It can be a tedious, and occasionally frustrating process; however, it is essential for the success of software development (Cheng & Atlee, 2007). In order for requirement elicitation to be successful, it needs close and effective interaction between customers and developers, so that customers' needs and wishes are understood and considered in the process (Seyff, Todoran, Caluser, Singer, & Glinz, 2015). It is no secret that the world of software development is rapidly changing, and close customer contact can be expensive and sometimes not feasible especially when many customers are geographically dispersed (Ali & Lai, 2016). As a result, companies need to adapt and move toward new channels of communication, mainly online-based, for capturing and analyzing customers' needs during the process of requirements elicitation (Ruhe, Nayebi, & Ebert, 2017). Some of these new channels include, but are not limited to, online forums, blogs, LinkedIn discussion groups, especially nowadays almost all application developers have a form of product feedback platform (Maalej, Kurtanović, Nabil, & Stanik, 2016).

It is thus fair to say that technology along with social networks are altering the norms of stakeholders involvement. Users' contributions to the idea generation of a new software product requirement elicitation process are beneficial and value-additive to software companies (Nambisan, 2002). Additionally, user involvement through professional online communities in product development is gaining increasing recognition in recent years and becoming the trend of our time. Users are more actively involved in the software development dialog with other users as well as developers and software companies are attentively listening (Romero & Molina, 2011). Companies are becoming increasingly geared toward focusing on the "crowd's" needs and inviting them to contribute innovative and new product ideas and users are often sharing their ideas and expressing their needs for new products freely via online communities. Companies are particularly attracted to this approach because it is considered a reflection of users' needs (Ruhe et al., 2017). "Crowdsourcing" is likely to enhance the quality and economic feasibility of gathering requirements, as well as allow the developers to gain a comprehensive up-to-date knowledge of how the system is fulfilling user's requirements and needs (Hosseini, Phalp, Taylor, & Ali, 2014; Groen et al., 2017). Thus, with better understanding of users' input and the proper development of user-requested products, companies could gain user acceptance as well as competitive advantage in the market (Bilgram, Brem, & Voigt, 2008).

Original ideas were found to be expressed by regular users as Magnusson, Matthing, & Kristensson (2003) discovered. User communities provide an efficient environment that is rich with innovative opportunities where users share a number of ideas that could be valuable during product development stages (Romero & Molina, 2011).

1.1 Data-driven RE approaches

Requirements are traditionally elicited, validated and analyzed using techniques such as workshops, interviews, as well as walkthroughs, which are all based on the co-presence of participating users. Due to their expensiveness in terms of cost and time especially when used with a large scale of users, innovative and possibly automated approaches such as crowdsourcing and text mining have been deemed necessary to solve these challenges (Ruhe et al., 2017). There is an increasing focus on online-based requirements engineering with text-mining based requirements engineering approaches and techniques that analyze the abundant user-generated information remotely (Hosseini et al., 2014). Simply worded, these text-mining (semi)automated approaches offer the potentiality to capture requirements, which people express in words, without the co-presence of those people or "crowd."

A number of data-driven approaches for requirements engineering have been proposed given the relatively inexpensive instruments for online data collecting. Some research is done using different terminologies referring to the same notion of text-mining user content for requirement elicitation purposes. Among the terminologies used to describe those approaches in the domain of requirement engineering are Data-driven RE (Maalej, Nayebi, Johann, & Ruhe, 2016), crowd-based RE (Sherief, Abdelmoez, Phalp, & Ali, 2015), Crowd centric (Snijders, Ozum, Brinkkemper, & Dalpiaz, 2015) and Crowdsourcing (Rouse, 2010). However, most of these approaches are focused on mobile app requirements engineering utilizing app stores user reviews (Carreno & Winbladh, 2013; Pagano & Maalej, 2013). There is little research, on the other hand, on collecting requirements for desktop applications through mining multiple data sources such as user forums and social media altogether in an attempt to collect and confirm features highly needed and requested. Non-mobile apps are usually discussed and reviewed through other channels such as forums, which are different from app store in terms of their format and types of topics discoursed. It is, therefore, advisable for companies that want to decide on which feature to implement in their next releases to explore and study user forums requests and select only the features that bring value not only to users but also producers.

1.2 Building Information Modelling (BIM)

The influence of technology advancements has impacted many disciplines and changed the way business is conducted. In the architecture engineering construction (AEC) sector, a widely known and used software type is **building information modelling, or BIM** (henceforth in this paper referred to as BIM). An example of a BIM tool is illustrated below in Fig. 1. In this context, BIM tools constitute an example of a desktop application that differs from the mobile apps that have been studied in most of previous research.

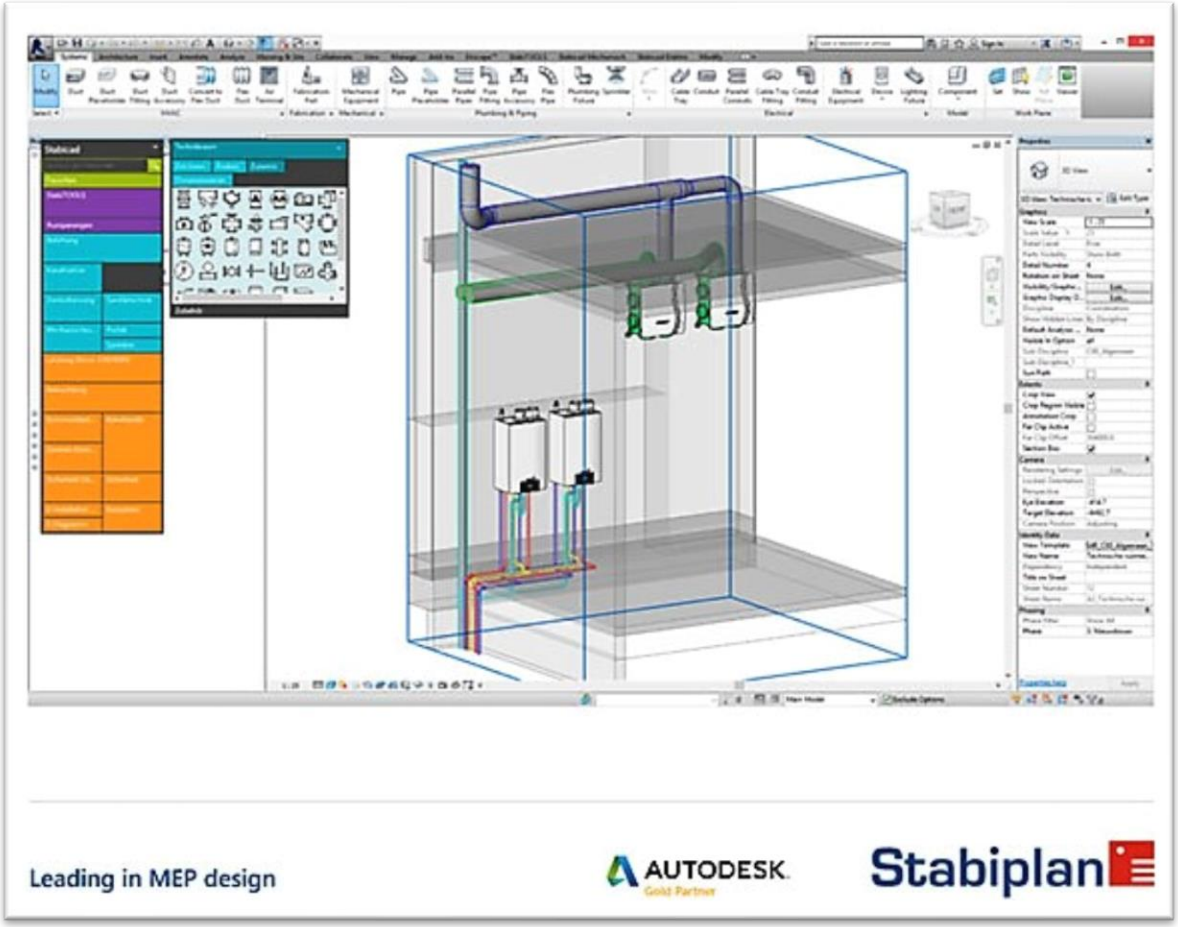

Fig. 1 BIM Tool Example from Autodesk Revit program (Photo source**:** www.stabiplan.com**)**

BIM has been used since the 2000s; however, its origins can be traced back to the early 1980s research of parametric modelling that was carried out in Europe (Azhar, Khalfan, & Maqsood, 2012). It has revolutionized the world of AEC and became the center of attention of building technologies as it stimulates incorporation of stakeholders' roles in a project as well as promotes

efficiency and optimization. According to the National Building Information Modeling Standards (NBIMS) committee of USA (as cited in (Azhar et al., 2012)), BIM can be defined by the following quote:

> "BIM is a digital representation of physical and functional characteristics of a facility. A BIM is a shared knowledge resource for information about a facility forming a reliable basis for decisions during its life cycle; defined as existing from earliest conception to demolition. A basic premise of BIM is collaboration by different stakeholders at different phases of the life cycle of a facility to insert, extract, update or modify information in the BIM to support and reflect the roles of that stakeholder."

Another definition of BIM is by Autodesk, a multinational construction, architecture and engineering software corporation. They define it as a 3D model-based process that supports the transparent collaboration between interdisciplinary professional teams to efficiently design, create and manage buildings throughout the entire construction project life cycle (Azhar et al., 2012). From the aforementioned definitions, it is fair to say that BIM is both a software and a process. Organizations and countries are in fact actively moving forward in the adoption and implementation of BIM in their construction laws and practices due to its reliability and effectiveness in achieving maximum results as well as cutting back on budget costs (Azhar et al., 2012). As a result, this field is getting the attention of software development enterprises that are compelled to continuously design and implement innovative ideas and solutions.

Autodesk Corporation is considered a leader in developing software for AEC industry worldwide. One of its widely used programs is Revit for designing mechanical, plumbing and electrical installations for buildings. Extra "apps" that can be added on top of Revit are being widely developed by several software companies specialized in BIM. These apps can be directly downloaded from the Autodesk Revit App store. An interesting feature of the Autodesk Corporation is that they established a user community forum called "Autodesk Knowledge Network[1]" specifically for users to voice their ideas and suggestions and ask questions regarding products offered. The forum is categorized by products and every product, such as Revit, has its own "Ideas" subsection for user-generated future features and enhancement recommendations directly related to that specific product. Mining those ideas and feature requests can be of great benefit to software companies especially those whose products are used by professional and technical users. The research project is performed, tested, and validated in collaboration with Stabiplan[2], a leading company in the Revit-based application developer specializing in mechanical,

[1] https://forums.autodesk.com/t5/custom/page/page-id/Ideas-Page
[2] http://www.stabiplan.com

electrical and plumbing building installations apps in Europe since 1990. Stabiplan is an established partner of the Autodesk Corporation in the field of architectural and constructional design and whose software products and apps are based on Autodesk Revit program. The results of using of this prototype by Stabiplan would indicate the perceived usefulness of this prototype and may be generalized to other software companies.

1.3 Problem Statement, Motivation and Research Questions

While there is a number of research work published for studying online communities, social media and forums in terms of their structure, threads solutions, threads traceability and so forth, (Baldwin, Martinez, & Penman, 2007; Sandor, Lagos, Vo, & Brun, 2016; Sondhi, Gupta, Zhai, & Hockenmaier, 2010; Wang, Kim, & Baldwin, 2012; Wanner, Ramm, & Keim, 2011**), there is yet not enough work focusing on requirement extraction from online user communities without the explicit involvement of stakeholders**. Some papers introduced an approach for stakeholders to elicit requirements with the active involvement of stakeholders in the process of elicitation and discussion. For instance, Castro-Herrera, Duan, Cleland-Huang, and Mobasher (2008), in their paper "*Using Data Mining and Recommender Systems to Facilitate Large-Scale, Open, and Inclusive Requirements Elicitation Processes*", introduce a process framework for eliciting requirements and needs from stakeholders involved in a project, pre-creates a number of discussion forums and then uses a recommendation system to assign each stakeholder to proper forum based on what they expressed, in order to engage them in a discussion with other stakeholders to ultimately arrived at final set of agreeable set of requirements. This approach is specifically appropriate when all of the stakeholders are known and reachable. However, this is entirely incompatible with the case for our research as the stakeholders, including the users, are unknown and impossible to reach and thus cannot be actively involved in the process.

Other research experimented with developing technical tools for the purpose of requirements extraction using app stores user reviews (Carreno & Winbladh, 2013; Chen, Lin, Hoi, Xiao, & Zhang, 2014; Guzman & Maalej, 2014; Maalej & Nabil, 2015). However, trying to apply them to professional software vendors with newly established app stores is not achievable due to the fact that their app stores are fairly new and therefore there is evident lack of appropriate amount of reviews. Nonetheless, many of these vendors have user forums that are used actively for a long time and, hence, rich of user ideas and feedback which makes forums a promising source for requirements extraction. Thus, they need a different approach to fit their case.

There exists some work on user forums analysis, as mentioned earlier, and the use of consumer "co-creation" in creating new products, yet there has been little research on requirement extraction from online user communities, or forums, especially professional user forums. Some studies focus on online communities in product development in the field of game development (Holstroem, 2001) and the study that investigates the identification of lead users in online communities in online communities for product development (Bilgram et al., 2008) other studies focused on distinguishing threads (Baldwin et al., 2007) and inspecting threads and analyzing their structure to determine whether a posted problem has received a solution (Wang et al., 2012) or automatically assess the overall quality of posts (Weimer, Gurevych, & Mühlhäuser, 2007).

User forums have different structure than the structure of app stores. Forums usually contain discussions about different topics, for example, with intertwined threads. Moreover, professional (technical) forums are structured to have users, as well as developers, reply to each other in the discussion thread. In app store reviews structure, however, replies to user reviews are not currently allowed. Users can only review the application without interacting with each other or the developers. It is also worth noting that these technical forums have slightly different commenting structure than that of the app stores in the fact that each comment is about a specific functionality of one or more apps (or sub-app) under specific platform, whereas the mobile app store reviews reference one single app. Thus, existing research about both topics has focused on the standard format of an app store reviews or forum threads and posts, yet little research focused on this specific type of forum structure, namely the work of (Kanchev, Murukannaiah, Chopra, & Sawyer, 2017). Thus, existing approaches cannot be directly applied to the case at hand and rather they could only be adapted and used as inspiration for a new approach to fulfill this study.

Additionally, insights about product requirements expressed by users in fora would be of considerable value for software companies as they reveal customers' needs. The Autodesk Forum, for example, is widely used site where professional and non-professional software users, including companies, and general public are communicating about BIM-related problems, enhancements, feature requests and new trends which could also represent an invaluable source of profitable future innovations. To the best of the authors' knowledge, no previous research has studied requirement elicitation from this type of technical forum in the past. Thus**, the aim of this research is to create a data-driven prototype to help software companies in the process of mining such online platform to discover requirements** such as feature requests posted as well as bug reports **and possible markets for them**, which can enable product managers make actionable decision to advance their products and expand their services.

It is lucrative to design a semi-automated data-driven approach whose primary objective is to build a prototype that can collect and analyze insights from a multimedia source that has the potential to

guide leading software companies in their journey of requirement engineering elicitation to develop innovative desktop-based applications. In order to explore the potentiality of discovering requirements from "Web 2.0" or user generated content site, the following main research question has been formulated:

"What are effective and useful data-driven approaches to elicit requirements and discover potential new markets for a technical software product focused on professional customers?"

To answer the main research question thoroughly, five sub-questions were formulated as follows:

RQ1: What are the existing studies done on eliciting requirements from user generated content on the Web?

This question aims to answer what existing studies on requirements elicitation are using user forums. In the literature review, an overview will be presented on studies and data-driven approaches that focused on requirements elicitation and extraction from user generated contents and their main findings.

RQ2: How to analyze professional user forum posts to identify candidate requirements for software products development?

For this question, the main source of data is the relative posts in Autodesk Knowledge Network forum where users express their suggestions and desires. A scraping tool will be used to scrape the website for relevant data which will be stored in a specific database.

RQ3: How to analyze Twitter to identify candidate requirements for software products development?

For this question, a preliminary scan of Twitter content will be evaluated based on its useful tweets (if any) that might carry relevant information about possible requirements related to BIM software in general, and Autodesk Revit in specific.

RQ4: What is the accuracy of the techniques developed to answer RQ2, RQ3?

To answer this question, the authors will perform a manual requirements extraction from the dataset (gold standard) and compare their results with the results statistical outcomes of the developed prototype.

RQ5: How to use information from Twitter and professional user forums to discover potential new markets?

This question will be answered based on combining the outcomes of question 2 and 3 with the assumption that geolocation data is available to integrate it with the prototype and to be further visualized in order to discover suitable markets for the requirements uncovered.

RQ6: To what extent are the outputs of RQ2, RQ3, and RQ4 perceived as useful by software product managers?

To answer this question, product managers from Stabiplan will be interviewed to evaluate the perceived usefulness of prototype through using it to accomplish some product management tasks and compare it with the original Autodesk user forum website.

1.4 Research Method

The aim of this research is to explore and find an alternative data-driven way to elicit requirements through the exploitation of user forums to assist product managers in their requirement elicitation tasks. Thus, the general research approach we use in this project consists of three steps: 1) a literature review where an understanding of the current existing body of knowledge regarding requirement elicitation using online sources is established, 2) artifact design where a prototype is built to help managers discover and visualize requirements from user fora, and 3) artifact evaluation where the prototype is evaluated using statistical measures as well as a series of expert interviews and surveys. Fig. 2 below provides a visualization of the general approach for this research, including the outcomes that each part yielded.

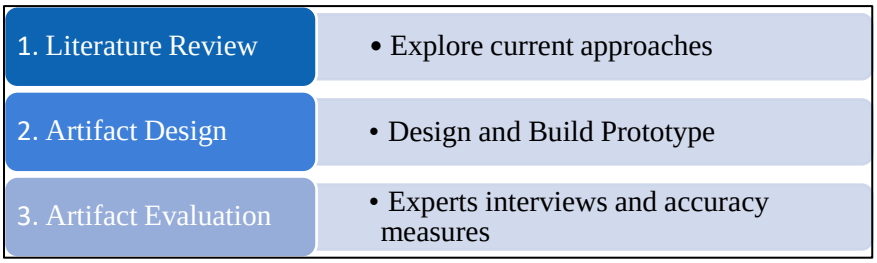

Fig. 2 Research Method

1. The first step is performing the literature review to explore and understand current works similar to the project and dissect their approaches and results. Further discussion of this step will not be discussed here since it is thoroughly explained in Chapter 2.
2. The second step performed is the artifact design during which we created a prototype for extracting possible requirements to: a) extract content from the user forum, b) process it using several natural language processing algorithms including Naïve Bayes Classification and LDA Topic Modelling, and c) Visualize the outcomes using visual analysis tools.
3. The final step is the artifact evaluation, which consisted of two parts of analysis: quantitative and qualitative as follows:
    a. For the quantitative analysis, our goal is to study how accurately the classification techniques can predict the types of the forum posts (bug report or a feature request). We evaluate the correctness and performance of the model using accuracy, precision, recall, and F-measure. This step was executed prior to the interviews as to ensure the prototype performed correctly. However, before we dive into the next sections, and to ensure no ambiguity in the terms used, there are a few statistical terms and measures that need to be briefly defined:

- Accuracy of the classification can be defined as the skill of the classification algorithm in predicting the correct category of the text. In other words, it is the percentage of the correct predictions divided by total predictions, or:

$$\text{Accuracy} = \frac{\text{correct predictions}}{total\ predictions} * 100$$

- Precision is the percentage of the predicted documents for a given category that are classified correctly, or:

$$\text{Precision} = \frac{\text{categories found and correct}}{total\ categories\ found}$$

- Recall (also known as sensitivity) is the percentage of the documents for a given category that are classified correctly, or:

$$\text{Recall} = \frac{categories\ found\ and\ correct}{total\ categories\ correct}$$

- F measure is the harmonic average of precision and recall (also known as F1 score):

$$\text{F measure} = \frac{2*Precision*recall}{precision+recall}$$

These three measures will be used to evaluate the accuracy of our classifier. Naive Bayes classifiers are known for their robustness and accurate results and have shown that they perform very well in text classification tasks (Nayebi et al, 2017a).

b. For the qualitative analysis, we evaluate the perceived usefulness of the prototype with expert interviews. For the expert interviews, we conducted an experiment to evaluate whether product managers find the prototype useful to ensure that this solution is satisfactory and feasible. During the experiment, the participants had to perform certain tasks using the prototype then fill in the perceived usefulness survey created by Davis (1989). Additionally, the participants had to repeat the same tasks using the Autodesk user forum then fill in the same survey with appropriate modification. To eliminate bias, the order in which the users used the supporting tools (Prototype or Forum) was shuffled across the participants so that when a user used the forum first then the prototype, the following user used the prototype first then the forum, and so on. Appendix B: Qualitative Evaluation contains the evaluation protocol with further details.

The remainder of this research is structured as follows: Chapter 2 provides a thorough literature review of related work and studies for requirement elicitation using online sources so far. In Chapter 3, the Artifact design is explained followed by the evaluation of the artifact in Chapter 4, including the design, qualitative and quantitative evaluations results. Chapter 5 will conclude the research and answers the research questions concisely. Finally, research limitations and future work are discussed in Chapter 0.

2 RELATED WORK

This section presents related literature regarding requirement engineering elicitation using data-driven approaches to extract knowledge from online user generated input in different platforms. This section also attempts to answer RQ1. There exists a number of studies conducted on extracting requirements from online platforms where users voice in their opinions and needs in the form of reviews. By analyzing online platforms such as user communities, social media, and app stores, the vast spectrum of information extracted can support requirements engineering decisions effectively. On the contrary, without such analysis of data, the developed software applications would be less wanted and useful by the users. This section starts with an overview of the literature protocol followed when conducting the related literature review, followed by discussions of literatures regarding the different platforms of interest to this research.

2.1 Literature Protocol

The search process for related work was conducted manually through websites such as Google Scholar, the ACM digital library, the IEEE library, Springer, Elsevier, science direct and many other well-known scientific papers repositories. Some of the Keywords used to find the articles include “requirements engineering in the world of apps”, “requirements engineering approaches”, “requirements elicitation”, “app store requirement engineering”, “data driven requirements engineering”, “requirements extraction in online communities”, “user forum requirement extraction”,” user reviews extraction”, “twitter mining” “twitter requirements engineering” “user forum information extraction”. The titles of the resulting articles were screened to find related and interesting articles which were later fully read and based on their research design, data used, and relevance to requirement elicitation or extraction, user-based requirement engineering, data-driven requirement engineering, and user reviews requirements extraction and whether it involved active participation of users and stakeholders. As in this research we do not involve users or stakeholders actively in the requirement extraction process, articles that required active user participation were eliminated. We also eliminated duplicate results and grey literature such as blog posts or articles published in non-scientific magazines. The snowballing technique was also applied to check for interesting citations found in an identified paper: we did search for the cited paper, read it to determine relevance, and considered the number of citations on Google Scholar to determine its popularity and credibility. An exception to the newly published papers is that new papers were

only included based on the reputation and expertise of their authors in the field of requirement engineering. This process was applied to every paper until the same references reappeared anew.

Several papers were excluded from the search results as they were considered outdated to be included in this rapidly developing field (more than 20 years old), broadly discussing RE (Nuseibeh & Easterbrook, 2000)) and its methodologies (Coughlan & Macredie, 2002) or non-web-based techniques such as the work of effectiveness of Requirements Elicitations tools by Davis, Dieste, Hickey, Juristo, & Moreno (2006), RE ecosystems and product life cycle (Knauss, Borici, Knauss, & Damian, 2012), or offering a web-based tool such as forums or surveys for stakeholders to elicit requirements including the active involvement of users and stakeholders in the process of elicitation and discussion (Castro-Herrera et al., 2008; Laurent & Cleland-Huang, 2009), as well as using stakeholder-based social network modelling (Lim & Finkelstein, 2012) and walkthroughs and interviews techniques which are considered irrelevant to this research as it seeks to discover requirements from user content on the web without the active participation of users or stakeholders. To elaborate, for instance, Castro-Herrera et al. (2008) in their paper titled "Using Data Mining and Recommender Systems to Facilitate Large-Scale, Open, and Inclusive Requirements Elicitation Processes" introduces a process framework for eliciting requirements and needs from the stakeholders involved in a project, pre-creates a number of discussion forums and then uses a recommendation system to assign each stakeholder to proper forum based on what they expressed, in order to engage them in a discussion with other stakeholders to ultimately arrived at final set of agreeable set of requirements. This approach is specifically appropriate when all of the stakeholders are known and reachable. However, this is incompatible with the case for our research as the stakeholders including the users are unknown and cannot be actively involved in the process. Similarly, papers that elicit requirements through the observation of anonymous users' behavior during interaction with a system and comparing it to a set of system assumptions are excluded such as the work by Brill and Knauss (2011), for example, as well as the work of Karlsen, Maiden, & Kerne (2009), and Mich, Anesi, & Berry (2005) which focused on creating requirements using creativity and art-based tools.

Papers included in this research were based on their direct link with the research objective and questions, on their recency, research design and techniques used. Below we present a thorough discussion of the related works included and their relationship to our research scope.

2.2 App stores

In previous research, user reviews have been the basis of studies with various purposes. Martin, Sarro, Jia, Zhang, & Harman (2017) provided an initial survey of literature between 2000 and 2015 that contains analysis of 45 studies on app store reviews, including a full section of Feature analysis studies, and it showed that in the last few years this area of study is becoming trendy and getting more research attention. The survey also includes a number of studies focused on extracting features from user reviews using different techniques such as natural language processing, topic modelling and clustering (Martin et al., 2017). Other studies focused on how to automatically filter, analyze, and classify reviews into predetermined categories for which they aimed to develop models and tools (Guzman & Maalej, 2014; Nayebi, Cho, Farrahi, & Ruhe, 2017). Carreno & Winbladh (2013) have developed an automatic topic extraction tool that helps developers and requirements engineers to analyze user comments in app stores and use it to adapt requirements for future releases. Such approaches are considered helpful and would complement our research approach. Some research is also focusing on data-driven approaches to extract requirements from publicly available data sources. Maalej et al. (2016) elaborate on and give an outline of the new trends for requirement engineering in the app-store domain. They emphasize the importance of actively including and interacting with all stakeholders and conclude that the process of RE should be more data-driven in the future.

Tools and techniques have been used to analyze the content of user reviews in domain of app stores. Chen et al. (2014) proposed a framework, AR-Miner, for app-review mining with analytical capabilities and lower human effort. The AR-Miner classifies reviews into informative and uninformative in five steps using topic modeling and was found to be more time efficient than manually extracting informative reviews. The authors performed the process by hand at first and measured the time needed, then used the AR-Miner tool and compared both times needed to achieve the goal. AR-Miner could be useful for this research combined with classification suggested by Maalej and Nabil (2015) to deeply analyze the data (Nayebi, Cho, et al., 2017). Maalej and Nabil (2015) introduced statistical techniques as well as NLP and sentiment analysis techniques on how to classify reviews into 4 different categories including feature request and bug report. Such techniques provide a good basis for this research approach especially when combined with sentiment analysis to categorize and sort the features extracted from the raw texts intended to be collected. Sentiment analysis has been studied to distinguish people's perspectives and feelings when writing tweets and reviews (Nayebi, Cho, et al., 2017). Generally, people's sentiment is reflected through their opinions, and today people tend to express their opinions about certain elements such as software products via Twitter or review platforms. Sentiment analysis is

advantageous in mining and analyzing social media text such as Tweets, and thus, might be useful for this research approach to study also users' sentiment regarding software products.

2.3 Online Forums

Online forums are online discussion platforms where users can discuss and exchange knowledge. Forums are popular text-based threaded communication websites and are typically domain specific such as medical forum, technical forum, media forums and so forth.

Forum mining has been investigated and recognized as an essential undertaking for various cases. The vast amounts of data piled up online represents an important knowledge base that can aid in different business activities such as customer support, software development and maintenance, and requirement engineering (Morales-Ramirez, Kifetew, & Perini, 2017). Mining forums content and user feedback could help identify trends, extract requirements, and create opportunities that have the potential to influence software application's success or failure (Hosseini et al., 2014), hence affecting business development decisions.

One interesting research line concerning forums is defining and distinguishing forum threads. The work of Baldwin et al. (2007) focused on the classification of Linux user forum threads based on the following characteristics: specificity or generality of the post, completeness of the original post, and whether or not the discussed problem is eventually resolved. Their research goal was to devise an advanced tool for information access and retrieval from forum threads. They developed an (automatic) system called ILIAD (Improved Linux Information Access by Data mining) which they trained using pre-annotated subset of threads having the aforementioned characteristics. They also performed different experiments with classification and regression machine learning algorithms. Though their extensive tests and experiments leading to positive results, the outcomes were not as good as they hoped due to disagreement in the annotation schema (low kappa) which caused the classifiers to perform below baseline. One of the reasons why this had happened according to the authors is that they converted the Likert scale they used to an ordinal scale. Nonetheless, this study contains valuable applicable methods and measures which could support our research, and their glitches should be taken into consideration when applying their approach on the case at hand.

Wang et al. (2012) have extended upon the work of Baldwin et al. (2007) and further explored the task of "Solvedness classification" using the ILIAD data set created by Baldwin et al. (2007). Solvedness classification refers to automatic prediction of whether a problem posted by a user in

a forum is solved or not through the inspection of thread structure and analysis of user replies. Their experiments were based on "stratified 10-fold cross-validation". The method they used is thread discourse structure parsing in "the form of a rooted directed acyclic graph over posts, with edges labelled with dialogue acts (DA)." Though this task was proved difficult to Baldwin et al. (2007), Wang et al. (2012) were able to achieve improvements in "Solvedness classification" accuracy with the use of gold standard discourse structure and their results surpass the baseline (Fig. 3). While their work is valuable, it is not directly related to our research approach as we are interested in collecting and extracting information (i.e. requirements) from users posts in the forum rather than trace and analyze posts structure and post replies relationship to answering a specific post question.

Feature Category	System/feature(s)	ACC_{gold}	ACC_{auto}
Baseline	ZeroR	.804	
	ADCS	.804	
DA-only	LastPostDA	.784	.780
	LastNonInitDA	.792	.788
	HasResolution	.804	.804
	LastPostDA +LastNonInitDA	.848*	.776
	LastPostDA +HasResolution	.864*	.780
	LastNonInitDA +HasResolution	.872*	.788
	AllDAFeat	**.884***	.776
LinkDA-based	LastPairDA	.832	**.816**
	LastSubthreadDA	.832	.792
	AllLinkDAFeat	.824	.792
	AllDAFeat +AllLinkDAFeat	.852*	.792

Table 4: Results over ILIAD, using discourse structure features from the gold-standard and also the discourse parsing model ("*" signifies a significantly better result than both baselines; the best result in each column is indicated in **boldface**).

Fig. 3 Results from Wang et al. (2012)

Sondhi et al. (2010) have studied information extraction from medical forum with the goal of extracting relevant sentences to a predefined set of semantic categories related to a medical case description. They used two supervised (machine) learning methods: Support Vector Machines (SVM) and Conditional Random Fields (CRF) in order to distinguish between sentences corresponding to medical problems and medical treatments. The results show that it is feasible to extract medical cases automatically from forums using the features proposed as they have proved able of improving the accuracy of extraction. Although this research focuses on medical contexts, it could be useful in this study in terms of providing an insight to the methods and algorithms it applies in distinguishing the sentences such as SVM and CRF.

A study by Sandor et al. (2016) has proposed a system that uses discourse analysis to detect types of sentences from user forum questions and classify them according to the information requests they contain into four categories: *anomaly*, *device property*, *explanation*, and *how to*'s. This is

aimed at helping (semi) automated answering systems find the right knowledge base that can be used to answer the user's question especially in customer care services. They used 150 random posts from a user forum online and established a golden standard in which they annotated the technical forum corpora manually and used it to find discourse patterns and identify type of sentences. They applied jointly a topic modelling method and TF-IDF to train their system along with bigram and part-of-speech features which they reported to have achieved the best results (see Fig. 4). Their results show that "discourse related features" are useful when working with complicated notions such as anomalies conveyed in question posts. Although the forum under study in this research has a different structure than our question-based forum, their method could be useful for this research.

		ANOMALY	HOWTO	PROPERTY	EXPLAN	NULL	AVERAGE
BIGRAM+POS	PREC	0.444	0.854	0.764	0	0.801	0.578
	REC	0.108	0.759	0.732	0	0.925	0.505
	F1	0.276	0.806	0.748	0	0.862	0.534
BIGRAM+POS+XIP	PREC	**0.5**	0.883	**0.854**	**1**	0.767	**0.8**
	REC	**0.27**	0.704	0.577	**0.5**	**0.971**	**0.604**
	F1	**0.385**	0.794	0.716	**0.75**	**0.869**	**0.703**

Fig. 4 Comparison of the classifier's performances used by Sandor et al. (2016)

There are several researches works that combine Natural Language Processing (NLP) techniques, classification techniques, text mining, and sentiments analysis to analyze the user feedback about software applications in online forums. However, there is only one research found by Morales-Ramirez et al. (2017) which used a linguistic technique called "speech act theory" in combination with sentiment analysis and NLP techniques to propose a method for analyzing and classifying user comments in online discussion forums. The goal of the proposed method is to provide an automated technique for discovering potential requirements contained in user discussions forums, namely user comments in issue tracking systems and open source software mailing lists. One of the findings of this research is that specific types of speech-acts can refer or hint to a possible requirement or bug reports. This work is beneficial in terms of providing such a tool that might help the current manuscript in processing user feedback and extract possible requirements sentences.

2.4 Social Media, Twitter

Social media nowadays is becoming an integral part of people's lives. Millions of users every day share every moment of their daily lives online. As a result, researchers geared their attention to study this phenomenon and analyze social media data for various purposes (Stieglitz, Brockmann, & Dang-Xuan, 2012). Twitter, Facebook, MySpace, Google+ and LinkedIn are few examples of popular social media sites that have been the basis of research studies. Twitter is considered one of the most used and influential microblogging platforms in the world as its users post short messages of no more than 280 characters called "tweets" on various topics daily (Atefeh & Khreich, 2015). The fast rate of which tweets are posted every moment and its popularity makes Twitter an attractive area for research studies such as predicting election votes and analyzing revolutions (Bruns & Burgess, 2011; Christensen, 2011; Stieglitz et al., 2012; Tumasjan, Sprenger, Sandner, & Welpe, 2010) natural disasters (Hughes & Palen, 2009; Mendoza, Poblete, & Castillo, 2010; Nayebi, Quapp, Ruhe, Marbouti, & Maurer, 2017) as well as daily life communications. An example of the usefulness of using Twitter in natural disasters and crisis is the case study of extracting information from tweets about the Fort McMurray wildfire emergency situation by (Nayebi, Quapp, et al., 2017). This study represents the usefulness and potential of extracting user feature requests and app information via Twitter analysis as well as raise awareness of the developers to respond to users' needs. The results have showed that the emergency apps currently available are missing about 80% of the features needed by users (Nayebi, Quapp, et al., 2017). Some tweets include users' opinions, reviews, discussions, and possibly recommendations of software products. As a result, researchers have been focusing on developing methods and tools for analyzing the content of tweets (Bruns & Stieglitz, 2013; Guzman, Alkadhi, & Seyff, 2017) in order to provide companies with means to gather insights to know what the general population is saying about their (or others') products. Further, these tweets might include information about possible software requirements essential for requirements engineers to consider or implement in their next software release (Guzman et al., 2017).

Asur and Huberman (2010) have studied how Twitter content can be utilized to predict future outcomes such as box-office movies revenues. Over a period of 3 months they have collected over 2.8 million tweets referring to about 24 movies. They focused on studying how buzz and attention is generated using Twitter as well as their effect on the movies performance in the real world and used linear regression model for predicting box-office revenues prior to movies' release. The accuracy of the results outperformed that of Hollywood Stock exchange (R^2= 0.97) (see Fig. 5). Additionally, Tweets containing links were found to help in publicizing movies, and movies with greater publicity outperform others in the box-office.

Predictor	Adjusted R^2	$p-value$
HSX timeseries + thcnt	0.95	4.495e-10
Tweet-rate timeseries + thnt	**0.97**	2.379e-11

TABLE VI
PREDICTION OF HSX END OF OPENING WEEKEND PRICE.

Predictor	Adjusted R^2	$p-value$
Avg Tweet-rate	0.79	8.39e-09
Avg Tweet-rate + thcnt	0.83	7.93e-09
Avg Tweet-rate + PNratio	0.92	4.31e-12
Tweet-rate timeseries	0.84	4.18e-06
Tweet-rate timeseries + thcnt	0.863	3.64e-06
Tweet-rate timeseries + PNratio	**0.94**	1.84e-08

TABLE VIII
PREDICTION OF SECOND WEEKEND BOX-OFFICE GROSS

Fig. 5 Asur and Huberman (2010)

Bougie, Starke, Storey, and German (2011) have studied the use of Twitter among developers of software engineering projects. The authors have used archival and qualitative analysis to quantify and manually analyze some Twitter use parameters in the developers' conversations including the topics discussed among developers such as Eclipse and Linux, number of directed messages sent from one user to the other and number of retweets and hashtags and compare them to the average Twitter users. They used free program called The Archivist with the help of Perl scripts to collect 11,679 tweets over a period of time and compared their findings to the work of Java, Song, Finin, and Tseng (2007). One of their findings is that conversations between developers in three identified communities have increased by 50% - 67% from 2007 Twitter study and the information sharing determined by URLs was also increasing by 6% - 24% (refer to Fig. 6 below).

Group	% @	% URLs
Twitter corpus collected by Java *et al.*	12.5	13
Linux June/July 2010	79.7	37.3
Linux January 2011	68.8	34.3
Eclipse June/July 2010	76.3	27.5
Eclipse January 2011	62.1	31.9
MXUnit June/July 2010	76.5	23.8
MXUnit January 2011	72.3	19.7

Table 2: The percentage of conversation (@) and information sharing (URLs) seen in the three communities at the two different time periods, compared to the 2007 Twitter corpus of Java *et al.*

Fig. 6 Bougie et al. (2011)

In their recent study of Twitter, Guzman et al. (2017) explored the use of Twitter as a communication tool and its relevance to software development stakeholders, especially requirements engineers. They focused their exploration on tweet content, usage, and possibility of automating tweet classification. Their dataset included over 6 million tweets representing 22 popular mobile and desktop applications selected by the authors. They used descriptive statistics to analyze the content of the tweets, namely content analysis techniques and manually identified the content categories of 1000 tweets and their relevance to stakeholder group (technical, non-technical, and general) and defined their sentiment on a Likert scale. For the tweet analysis automation potential, the authors exploited machine learning and lexical sentiment analysis techniques such as SVM and decision trees classifiers on a dataset of 10,986,495 tweets about 30 different software applications. When comparing the two techniques, their analysis results confirmed that SVM performed better than Decision trees. The main findings of their work are that relevant information about requirements engineering and software development do exist in tweets, and automation of this sort of Twitter content exploration is crucial for informing requirements engineering and software development activities. Fig. 7 sums up this result. Additionally, the lexical sentiment analysis used for extracting tweet sentiments automatically has yielded a strong correlation (Spearman's Rho coefficient = 0.60) to human judgement when analyzing tweets that are human generated and URL-free.

Table 7 Relevance classification results

	Technical				Non-technical				General public			
	Precision	Recall	F_1	F_5	Precision	Recall	F_1	F_5	Precision	Recall	F_1	F_5
Naive Bayes	0.38	0.80	0.52	0.77	0.82	0.67	0.74	0.67	0.82	0.68	0.74	0.68
Multinomial NB	0.30	0.84	0.44	0.79	0.69	0.82	0.75	0.81	0.69	0.82	0.75	0.81
SVM	0.54	0.44	0.48	0.44	0.74	0.77	0.75	0.77	0.74	0.76	0.75	0.76
J48	0.50	0.30	0.38	0.30	0.77	0.73	0.75	0.73	0.78	0.74	0.76	0.74
Random forest	0.73	0.24	0.36	0.25	0.79	0.74	0.76	0.74	0.80	0.74	0.77	0.74

Fig. 7 Guzman et al. (2017) classification results

In the recent study by Nayebi, Quapp, et al. (2017), it appears that Twitter can provide integral information for mobile app development. In their study they explored the existence of user feedback about apps in sources other than app stores reviews such as in a social media source (i.e. Twitter) and its potential in acting as a complementary information source of providing support for app developers. They studied the correlation between the number of tweets and app reviews and the alignment of sentiments between app reviews and tweets by analyzing Twitter content of about 30,793 apps in a period of 6 weeks. They used Automatic classification with SVM and Naive

Bayes to classify the content of tweets and reviews into fine-grained categories, then applied topic modeling on every class and defined two categories only: Feature request and bug report. Their results show that they were able to collect 22.4% more feature requests from twitter than app store reviews alone. Fig. 8 illustrates their results while their NLP process is depicted in Fig. 9 below.

TABLE I
PRECISION, RECALL, AND F-SCORE (F1) OF NAIVE BAYES AND SVM CLASSIFIERS SEPARATELY FOR TWEETS AND FOR REVIEWS.

Reviews

Classifier	Feature request			Bug report		
	Precision	Recall	F1	Precision	Recall	F1
Naive Bayes	0.89	0.81	0.84	0.91	0.82	0.81
SVM	0.84	0.79	0.81	0.78	0.73	0.75

Tweets

Classifier	Feature request			Bug report		
	Precision	Recall	F1	Precision	Recall	F1
Naive Bayes	0.76	0.61	0.67	0.77	0.70	0.73
SVM	0.56	0.50	0.52	0.61	0.54	0.57

Fig. 8 A summary of their classifiers' precision, recall and F-score (Nayebi et al., 2017a)

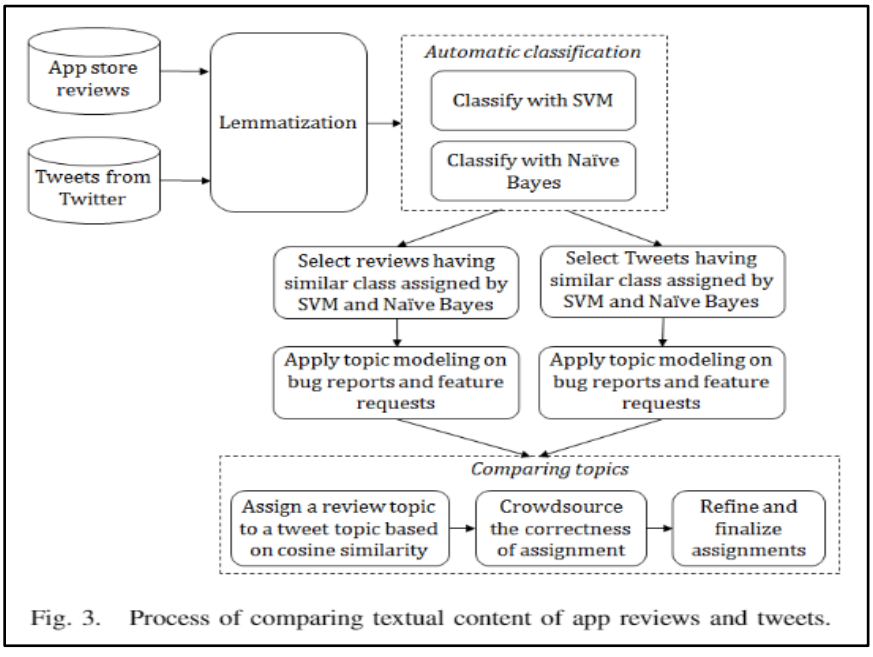

Fig. 3. Process of comparing textual content of app reviews and tweets.

Fig. 9 NLP process used by Nayebi et al.(2017a)

Although Twitter is regarded as a rich social media source of information by many researchers in the field of mining applications reviews, as discussed above, there were scarce tweets regarding a highly technical software such as Revit. During a preliminary data search in Twitter for keywords such as “Revit bug/problem”, “Revit function/requirements”, “Revit fix”, “Autodesk fix”, “Autodesk ideas”, “Autodesk problem/suggestion”, “Revit not working”, “Autodesk annoying” and many similar queries resulted in less than 10 obsolete tweets (dated since 2011 till 2014) that contained conference announcement, BIM events, and or advertisements for certain products or updates. However, none of the queries resulted in any relevant tweets. This could be due to the fact that highly technical applications usually well supported by their communities and have their own forums and discussion outlets which leads few people to use twitter for discussing and/or reporting issues and suggesting ideas. Tweets found were mainly outdated and mostly contained announcements about the release of newer versions of the software or declarations of relevant technical events or conferences.

2.5 Visual Requirements Analytics

The emerging field of visual analytics an opportunity to give information a clearer meaning and image. It was coined by Thomas and Cook in 2005 when he published his book “Illuminating the Path.” It is defined as “the science of analytical reasoning using interactive interfaces” (Thomas & Cook, 2005). Visual analytics can simplify complex problems that require massive human and machine analysis capabilities and enable effective human-information dialogue through its visual representations. With the unprecedented rate at which data is generated and collected, discovering and extracting requirements can be a burden to humans to process and analyze. However, with visual analytics the ability to analyze requirements and take actionable decisions at a lower cost makes it feasible and more attractive to decision makers (Reddivari, Rad, Bhowmik, Cain, & Niu, 2014). By combining the effectiveness of machines with user knowledge and creativity, visual analytics encourage decision makers to interact directly with the data visualized, understand it, analyze it, and discover knowledge within it that aid in solving complex business problems (Maalej et al., 2016). Fig. 10 below illustrates this process.

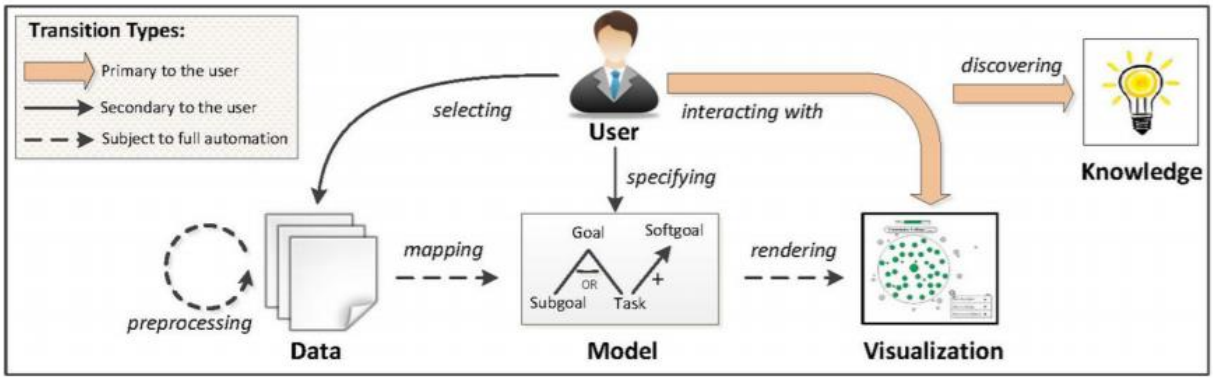

Fig. 10 Visual analytics process framework (from Reddivari et al. (2014))

There exist various research publications in the domain of visual analytics that emphasize its concepts, use, and importance. To name a few, there is the work of Keim, Andrienko, Fekete, Görg, Kohlhammer, and Melançon (2008) who regard visual analytics "an integral approach to decision-making, combining visualization, human factors and data analysis" and the contributions of Keim, Kohlhammer, Ellis, and Mansmann (2010), Thomas and Cook (2005). In the visual requirements analytics where the focus is on applying visual analytics to requirements engineering there are the works of Reddivari et al. (2014), Cooper, Lee, Gandhi, and Gotel, (2009), Gandhi and Lee, (2007), Reddivari, Chen, and Niu (2012) and many more. This is to show that Visual analytics plays an essential role in providing robustness and effective means in requirements engineering tasks. Therefore, visual requirement analytics provides an interactive means for stakeholders to understand, analyze, and interact with requirements and be able to extract information directly from the visualizations, which in turn, makes the process of discovering requirements for desktop software apps more efficient and effective.

In summary, it is complementary for requirements discovery and data analysis research to include widely used sources such as user fora for insights regarding software development and exploit the visual analysis tools available nowadays. Online users often post about their sentiment regarding a product, report bugs or request features not only through product or app review portals but also through social media websites. Organizations try to actively participate in social media sites to adapt to this fast pace technology advancements as well as to elicit requirements from users (Castro-Herrera et al., 2008). This also makes it important for companies to constantly absorb user feedback and make efforts to grant these requests and implements them in their products. Sometimes redundant features are requested, or the company ends up eliciting more new features than needed for a specific product. It is crucial for organizations to conduct online data analyses to understand customer needs and identify value adding feature requests as well as non-value

adding feature requests which can help avoid unnecessary costs due to over-implementing features. As seen from the literature review, a data-driven approach that combines Autodesk user forum information, a highly technical and professional user forum, with natural language processing and visual analytics tools, to analyze and extract requirements for developing desktop program apps has not been explored in the general body of RE studies.

3 ARTIFACT DESIGN

In this chapter, we discuss the design of our prototype. We aim at utilizing text mining techniques to analyze what people are posting on a technical user forum. More specifically, our goal is to study how accurately the classification techniques can predict the two types, bug report or a feature request, in the text forum posts. Additionally, we propose a visualization prototype to provide an appropriate visualization for the analyzed data, and ultimately support software product managers with requirements. This approach can be used by product managers in software companies to semi-automatically analyze possible data sources and extract a list of potential requirements as well as discover suitable markets to guide their product development decisions.

For this research, we limited the available and relevant data to the Autodesk Knowledge network user forum as it is a source of information with professional and specialized users. Based on literature study and manual preliminary inspection of some Autodesk related user forums such as Revit forums, Revit MEP, AUGI Forum, and after interviewing few experts in the field, we identified this platform as having a rich content and wide-use as well as being very useful source of information for requirements extraction and potential new markets discovery for Stabicad, the software application based on Revit produced by Stabiplan. Fig. 11 provides a visualization for the overall approach for the artifact design and evaluation:

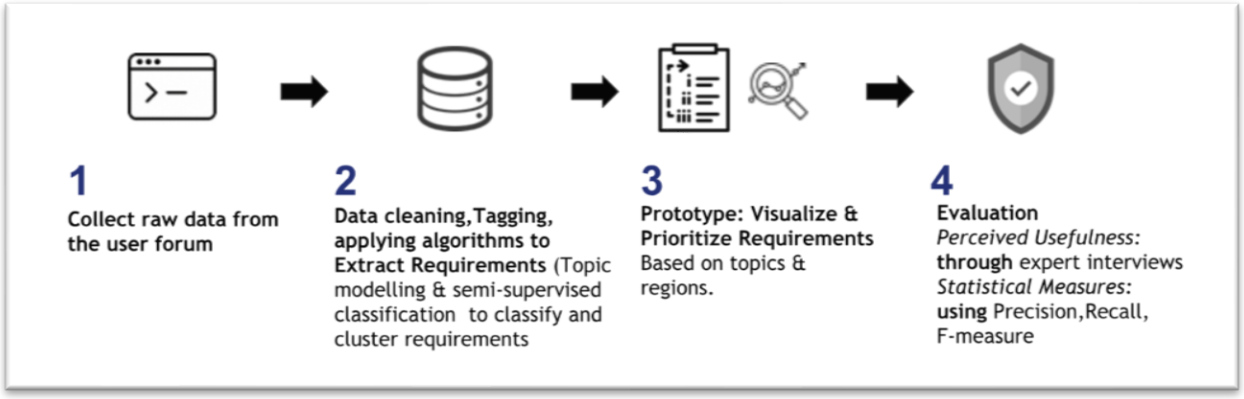

Fig. 11 Artefact Design and Evaluation

1) To scrape the data, first the structure of the websites is analyzed, and downloadable parts need be identified (e.g., links, user profile link, number of likes or votes etc.). Then, a data structure is devised to clearly define the metadata to be extracted from each source (e.g.,

title of post, content of post, other data, user name, user location, etc.). A data crawler will be used to scrape data from Autodesk forum.

2) Once the data is collected, a linguistics analysis to classify and cluster the data will be conducted using Natural Language Tool Kit 3 (NLTK) software package as well as Topic Modelling using the Latent Dirichlet Allocation (LDA) algorithm to further identify requirements, classify them in terms of features requests, bug reports, and cluster them into topics. The accuracy results from this analysis step will answer RQ4. The requested features extracted could provide insights regarding unrefined requirements and need to be further explored by product managers.

3) Using a visual analytics tool, a dashboard is created to visualize and analyze requirements and to prioritize them based on their popularity and geographical origin. This step will help answer RQ5.

4) Finally, interviews with experts will be conducted to evaluate the perceived usefulness of the prototype versus the regular forum using a modified version of the perceived usefulness survey by (Davis, 1989) which will answer RQ6

User posts in this forum either: a) describe a problem or limitation of Revit, or b) suggest a new idea for Revit. A post in this forum is a one level thread consists of a title, description, with date, timestamps, and the post status automatically generated. Any user can reply to a post in a chat-like way (see Fig. 12 below) and vote it up. Voting down is not allowed.

[3] http://www.nltk.org/

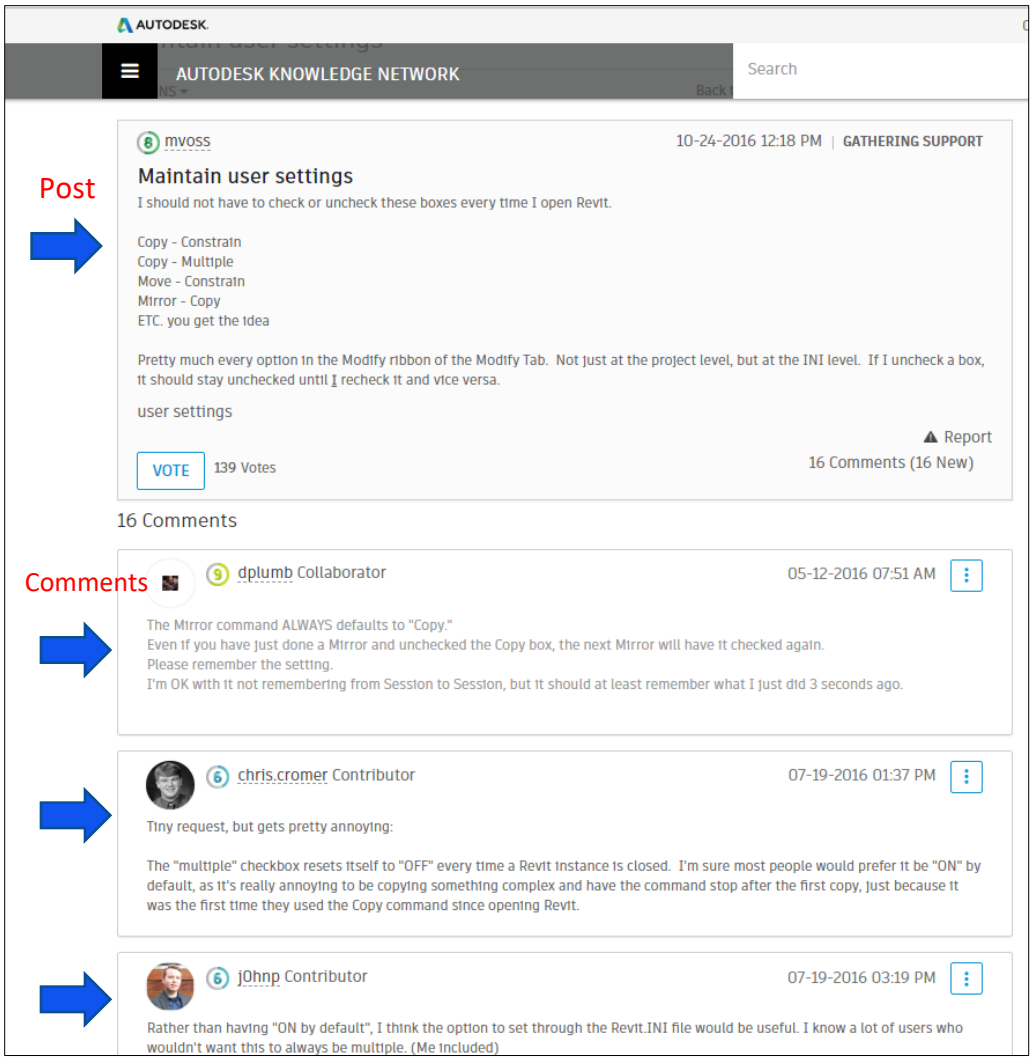

Fig. 12 Screenshot of Autodesk User Forum showing post format and user replies

In the next subsections we elaborate on each step of making our prototype and the different tools we use to achieve each step.

3.1 Data Preparation (Step 1)

The data collected from Autodesk Idea Forum included posts from mechanical, electrical and plumbing categories. Data crawled using a free open source software called Scrapy[4] over a period of two months (from January 15 to February 28, 2018) and **totaled 1,378 posts** among which were 512 mechanical, 505 electrical, and 361 plumbing; posts containing non-English content were

[4] https://scrapy.org/

excluded. After much scrutiny, comments on posts were ruled out of the research due to two main reasons. First, most comments contained either a question about the idea of the post or insults to the author of the idea. Second, some comments contained images and no text which made them invalid for the analysis. All data entries were in text format and saved in separate excel files based on their category (an Excel file for all Mechanical posts, and an Excel file for all Electrical posts, and an Excel file for all Plumbing posts).

For the **gold standard**, a sample of 121 files was randomly selected from all three categories (mechanical, electrical and plumbing) and manually classified by a total of three coders. The three coders were given a detailed explanation of classification process beforehand and followed a thorough classification protocol with guiding examples to help them with the process (See appendix A). An intercoder reliability test was calculated using SPSS resulting in a **Cronbach's Alpha** of **0.831**, thus, a high agreement between the three coders. Further, the tagged data was divided into two balanced subsets for each class type (61 samples manually classified as Feature request and 60 samples as Bug report). The gold standard data was then mixed and randomly split to use 70% for training and 30% for testing the classifier. For each data entry, only text content was used to construct a corpus to be used by the classification program; thus, only the title and description contents were stored in a text file format as they contain informative information about the post since few posts contained (long) titles only and a blank description field.

3.2 Data Cleaning, Classifying and Topic Modelling (Step 2)

Removing the noise and unnecessary words from texts is a common practice in the NLP process. To clean our data, we performed the following steps:

A. **Stop words:** words such as 'the', 'and', 'too', 'I', 'Hi', 'Revit', 'Autodesk' etc. were removed from all texts to eliminate noise and help improve the accuracy of the classifier. However, words that could bare information helpful to the classifier such as 'should', 'bug', 'add' were kept unremoved.

B. **Lemmatization**: Lemmatization describes the original root form of a word. All words in the data were converted to their root word. For example, 'working' becomes 'work', 'better' and 'best' become 'good'.

C. **Lowercase**: all words in the text were converted to small letter cases so that JOB and job are treated as one word by the classifier.

These steps are considered part of data preprocessing and can help the classifiers increase their performance and accuracy (Maalej & Nabil, 2015).

As a part of our artifact design, we want to *classify* the posts into either *bug report* or *feature request*. We assume that when users write a post, they tend to use similar linguistic patterns for reporting a bug or requesting a feature. For example, the post "This should be fixed in the next version" can be understood by the developer or product manager as a bug report and asks for a fix. Similarly, if the user writes something like "Add a close button to the view window", the developer can understand that this is a feature request.

However, in order to use the best suitable algorithm for our study we need to assess the effectiveness of different classification algorithms in automatically classifying the forum posts. The subject of classifying text as a bug report or a feature request is not new and has been explored for many years in research work especially classifying app stores reviews and Twitter tweets. Despite the ample studies published in this topic, this task is still not trivial and requires extensive capabilities (Groen et al., 2017; Maalej & Nabil, 2015; Snijders, Ozum, Brinkkemper, & Dalpiaz, 2015; Wang, Kim, & Baldwin, 2012; Weimer, Gurevych, & Mühlhäuser, 2007; Maalej et al., 2016). Since we are studying a user forum and not app stores, our approach to selecting a classifier is to evaluate the most used classifiers in the literature and use the one with best average accuracy results.

The most trivial way of classifying a document is to check it for the presence of certain keywords, or word matching (Maalej & Nabil, 2015). We used the list of keywords indicating a bug report or a feature request created by Maalej and Nabil (2015) as we observed the similarity of the words in this table and in the forum posts when performing preliminary scanning of the posts. Additionally, we observed few extra words used commonly when requesting a feature or reporting a bug so we added them to the list. We then wrote a program that checks the presence of any of the keywords in the post and determine if it a bug or a feature request and label it accordingly. This added measure is to help compare this simple classification technique's accuracy with other algorithms as well. The compiled list of keywords used is show in Table below.

Table 1 Keywords used by indicate bug or feature request (Adapted from Maalej & Nabil,2015)

Class Type	Keywords
Feature Request	Idea, add, please, could, would, should, hope, improve, miss, need, prefer, request, suggest, want, wish, implement, give
Bug Report	bug, fix, problem, issue, defect, crash, solve, sucks, workaround

Although many studies have used *Naive Bayes classifiers* in similar research endeavors due to its robustness and accuracy with a small training set and less training time than other classifiers (Maalej et al., 2016; Guzman & Maalej, 2014; Maalej & Nabil, 2015; Snijders et al., 2015), we test three classifiers among the widely used classifiers in literature and validate them using 10-fold cross validation algorithm to select one of them in our research (namely we compare the performance of Naive Bayes, Support Vector Machine and Bernoulli Naive Bayes). It is considered common practice when performing a supervised machine learning to test and validate a model using a reserved part of the truth set data. For this model validation, we reserved a part of the truth set to used is with Cross validation technique. Basically, cross validation is to perform multiple evaluations on different test subsets, then combine the scores from those evaluations into one average score. More specifically, the data is subdivided into 10 subsets called folds. For each of these folds, the model is trained using all of the data except the data in that fold, and then the model is tested on that remaining fold. Although the individual folds might be too small to give accurate evaluation scores on their own, the combined evaluation score is based on a larger amount of data, and is therefore considered reliable (Buitinck et al., 2013). After several repetitions of our 10-fold cross validation program, we calculated their average accuracies (**Table 2** Below), and we selected Naïve Bayes classifier to be applied in our research as it achieved higher accuracy for the data used; we used the golden standard data to perform this step.

Table 2 Accuracy of three classification algorithms during 10-fold cross validation

Fold	Naive Bayes	Support Vector Machine	Bernoulli Naive Bayes
1	0.94117	0.7532467	0.785714
2	0.92592	0.764705	0.825396
3	0.93023	0.792682	0.80
4	0.93589	0.764705	0.8045977

5	0.93023	0.761194	0.804878
6	0.93023	0.758620	0.789473
7	0.91764	0.738095	0.8196721
8	0.92307	0.783783	0.809523
9	0.94117	0.80	0.772727
10	0.92	0.771084	0.8139534
Avg. Accuracy	**0.92949**	0.7688119	0.8025937

In natural language processing, *Topic modelling* is an unsupervised learning technique whereby it uses models such as LDA to cluster subsets of text corpora based on some notion of similarity. LDA is a "generative probabilistic" model used to infer topics from a collection of text documents (Beli, Ng & Jordan, 2003). It represents documents as vectors of word counts (bag of words) and combines it with a set of parameters including a clustering parameter called "number of topics", all of which are then used by a function that tries to learn how documents are generated and then clusters them accordingly. This technique is widely used in RE research to aid practitioners in clustering requirements (Duan, 2008). For example, it was used to group features into meaningful high-level themes (Guzman & Maalej, 2014) and validate requirements topics traceability (Hindle, Bird, Zimmermann, & Nagappan, 2012), as well as automatic generation of requirements from connecting similar ideas from creativity workshops content (Bhowmik, Niu, Mahmoud, & Savolainen, 2014).

One of the most important parameters for LDA models is the number of topics. In order to determine the best LDA model to use, a GridSearch was performed to determine which number of topics is optimal for the model. To speed up the learning algorithm of the model, it is recommended to gradually reduce the learning rate, also called learning rate decay, over time in order for the algorithm to converge (Beli, Ng & Jordan, 2003). Basically, the grid search constructs multiple LDA models for all possible combinations of different topic numbers and learning rate decay values. The resulting best LDA model appears to be at number of topics 10. Thus, for our case 10 topics were ideal to use for our LDA model. However, for different data sets, the grid search should be repeated in order to yield an appropriate number of topics to be used for the specific research case at hand. Fig. 13 below illustrates the result graph of the grid search.

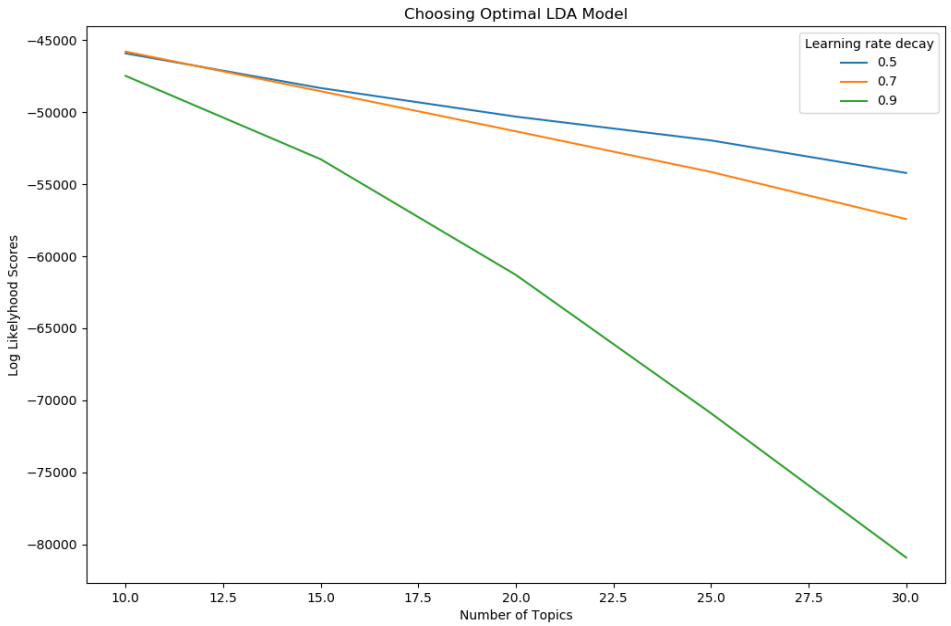

Fig. 13 Choosing the best LDA model

For simplicity and to better analyze the underlying possible topics of our data, and since all steps for applying topic modelling on the other two categories, Mechanical and Electrical, are identical, we chose to focus on one data category namely Plumbing. Therefore, topic modeling is applied to cluster posts that share a familiar theme into 10 possible topics. The algorithm allows us to generate top words that appear to have higher probability to occur in each topic are used to label the individual topics. We use these topics to differentiate the different clusters of posts during the analysis with the visualization prototype. The visualization prototype is discussed more elaborately in the following section.

3.3 Visualization Prototype (Step 3)

As a final step of the artifact design, a visualization of the classified and clustered data is created in the form of a dashboard using a commercial software called Tableau[5]. The reason why we use this software is because it is accessible, robust, and can support the visualization and analysis of mixed data types and structure in a meaningful and simple manner that other tools fail at. This visualization prototype represents our proposed concept of how to analyze and visualize requirements for Revit forum data which allows product managers to interactively analyze the outcomes and, consequently, gain insights to make decisions. Fig. 14 below is a screenshot of the

[5] www.tableau.com

prototype dashboard. The dashboard prototype offers various possible representations of the classified data with an emphasis on the geographical location. This is because the case company values the location data as it provides them with a way to strategically fulfill users' needs of new functionalities and thus decide the next destination for their future applications. Many forums are lacking this feature and managers must dig deeper in users' profiles to extract geographical locations; with this prototype and its visual representation this issue is solved.

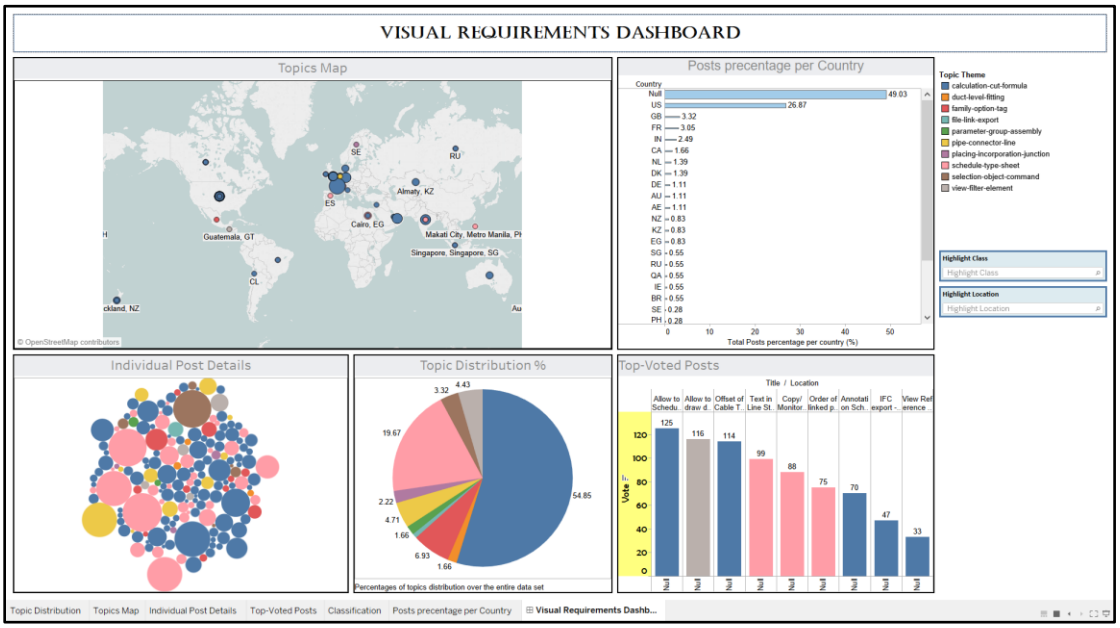

Fig. 14 Dashboard prototype for visualizing requirements (plumbing data sample)

The above dashboard prototype visualizes the data in five different ways so that the product managers can interactively analyze the extracted data efficiently. Colors in the dashboard are used to differentiate topics from one another. Each part of the dashboard will be explained below in details:

1. *Topics map* (Fig. 15) provides a geographical view of the physical locations of each individual post (and hence user's location). Giving product managers with an instant view of which region requires their attention and allows them to focus more on fixing issues per region.

Fig. 15 Map view of posts

The size of each bubble is representative of the amount of posts produced in that region, and the color indicates the topic to which the posts belong. This part of the prototype will additionally help to discover markets thus answering RQ5.

2. *The individual posts details* are represented by the clustered bubbles. Each bubble contains details of a specific post including its title and description, date, class (bug or feature), topic theme, number of votes, and country (see Fig. 16 below). Note how the size of each bubble is dependent on the number of votes (or likes) each post gets from the Revit community.

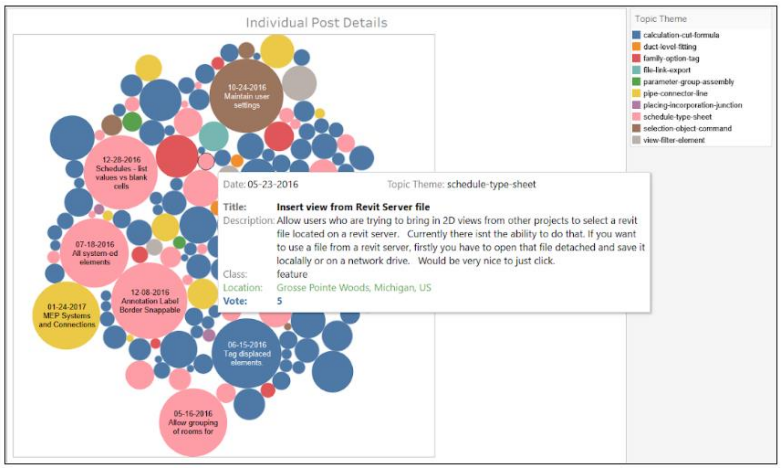

Fig. 16 Individual posts details

3. *The topic distribution percentage* is represented by a pie chart showing the division of topics among the analyzed posts. Each pie piece is labeled with the its corresponding percentage and hovering over any pie piece specific details will appear.

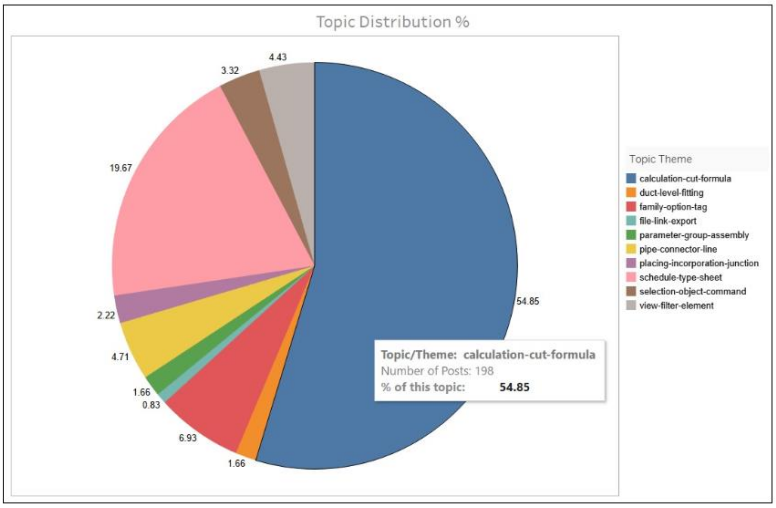

Fig. 17 Post distribution percentages

4. *The post percentage per country view* contains a ranking of the active country in the forum. Basically, for each country present in the data the tool calculates the percentage of its posts with regards to the total number of posts. Note that in our data almost half of the users do not reveal their location data in their profiles, therefore their country value is null.

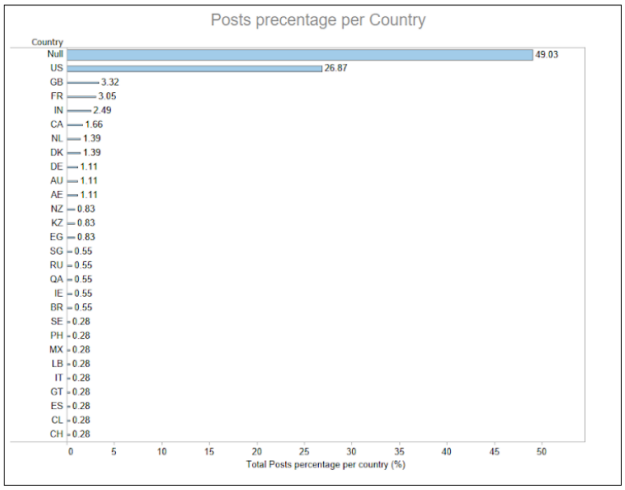

Fig. 18 Posts percentage per country

5. *The top voted posts* are depicted in an interactive colored bar chart graph (Fig. 19) showing the topic to which each post belongs, its locations, and total value of votes. By hovering over any bar, the tool reveals specific details of the post it represents.

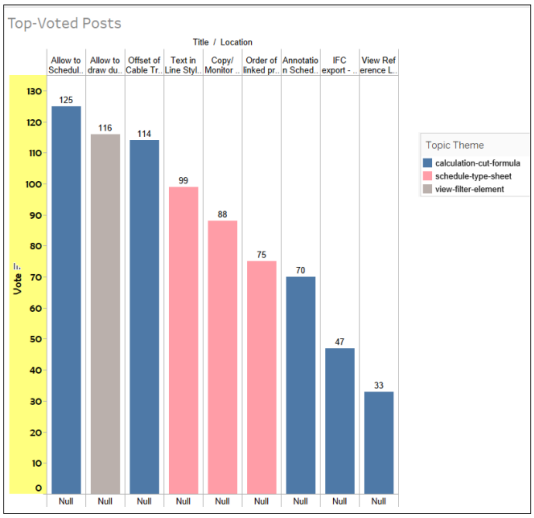

Fig. 19 Top voted posts

Additionally, by clicking on any item in the dashboard, the related information is filtered out in the five views as in Figure 20. Thus, it can assist product managers and provide better visualization and analysis of requirements which leads to effective requirements discovery and extraction.

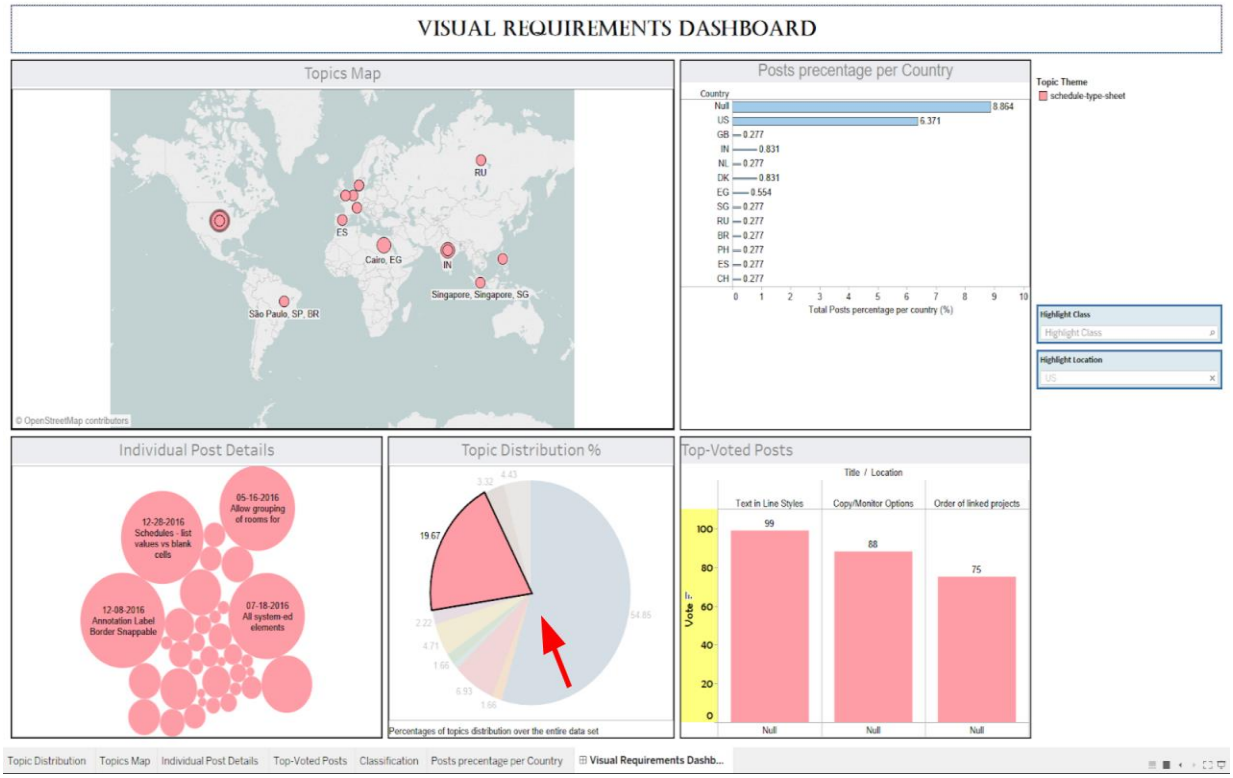

Fig. 20 Interactive filtering of one topic (*Figure Explanation: this view is triggered by clicking on a topic in the distribution pie chart (one pie piece) which filters out the related posts in the map and thus showing their exact location, separated those topic related posts in the details view, and ranks them based on their votes in the Top voted post view*)

4 EVALUATION

This section provides the evaluation measures we followed in order to evaluate our prototype as well as a detailed analysis of part of the results obtained.

4.1 Design

Our method validation process consisted of two goals, to assess: a) accuracy of some commonly used classification algorithms and use the most accurate for our prototype, and b) the perceived usefulness of the prototype.

First, we measure the accuracy of Naïve Bayes classifier statistically using Precision, Recall, and F-measure (defined in section 1.4). Precision in this context of classification is the number posts correctly labeled (true positive) divided by the total number of posts labeled correctly and incorrectly (false positive) as belonging to a class. Whereas recall is the number of true positives divided by the total number of posts that belong to that class (that is, sum of true positives and false negatives). F-measure is the harmonic average of precision and recall and is an indication of the confidence level. This step will answer RQ4.

Second, the qualitative evaluation step. This step is to answer RQ6. Before we proceed to the evaluation, we need to provide some context for the experiment:

Goal: to evaluate the perceived usefulness of the prototype through interviews with experts from the host company, Stabiplan,

Subjects: the subjects of the experiment were randomly asked to volunteer in the experiment. The six participants who volunteered consisted of five product managers and one support engineer agreed to participate, among whom was one female. The average years of experience among the subjects ranged between 1.5 to 4 years of experience. The total duration of each interview ranged between 30-45 minutes.

Procedure: The experiment was carried out in the Stabiplan headquarters. Each subject was interviewed individually in a meeting room. We started by explaining the evaluation protocol to evaluate the prototype. Next, the tasks they needed to perform were also explained. Then a through explanation of the prototype and how it works as well as instructions on how to navigate the dashboard were given. They were also given a detailed explanation of Autodesk forum structure and functionalities. This was followed by a walkthrough of the survey, during which all questions

were explained to the experts. The experts were then asked to provide their feedback on the prototype via the use of online survey.

4.2 Analysis

The following sections present the analysis of the aggregated findings from the NLP process and the evaluation experiment we conducted. The results of the classification and unsupervised clustering are reported in the following subsections as well as the results of the perceived usefulness experiment. Keep in mind the limited size of our data and experiment when reading the results.

4.2.1 ACCURACY OF NAÏVE BAYES CLASSIFIER

We trained the Naïve Bayes Classifier using the entire labeled dataset, which included 121 posts according to methods specified in section 3.2. We estimated the average accuracy of this classifier to be **89%**. To further corroborate these results, a fresh sample (on which the classifier has not been trained nor tested) of 45 pre-labeled posts was used to validate the classification model. The model classified 28 correct labels out of the 45, thus, an average of **62% accuracy**. Given the small data set this research had, the accuracy result is considered acceptable especially when compared to the comparable work on mobile app reviews of Maalej and Nabil (2015). Table shows the results of the Naïve Bayes classifier.

Table 3 Results of simple keyword matching and Naive Bayes Classifier (based on the 10-fold cross validation for NB)

Type of classification	***Type/class of post***	***Ave. Precision***	***Ave. Recall***	***Ave. F-measure***
Naive Bayes Classifier	*Bug Report*	0.9736	0.9291	0.9508
	Feature Request	0.9365	0.9762	0.9558
Simple keyword matching	*Bug Report*	0.5	0.25	0.333
	Feature Request	0.505	0.754	0.605

For feature requests, the precision reported by the algorithm is 93%, and the recall value is up to 97%. This means that for feature request the classifier labels relevant sentences accurately,

and also labels extra sentences as feature request that might not belong to that category. For the bug report, the precision reported by the algorithm is 97%, and the recall value is 92%. This means that for bug report the classifier labels relevant sentences accurately, but not all sentences that might not belong to that category are detected. In literature, accuracy of this type of classifiers in the range of **60-90%** is considered reasonable (Maalej et al., 2016; Maalej & Nabil, 2015; Nayebi, Cho, Farrahi, & Ruhe, 2017).

4.2.2 TOPIC MODELING RESULTS

The results of the evaluation were further analyzed using LDA to identify common topics across the forum posts and the results of this analysis are reported in this section. The results of topic modeling consist of the different groups of topics discovered in the documents, as well as common words per topic, the distribution of topics over the entire documents.

As for topics, Topic 5 talking over "View" "Filter" and "Element" seems to be the popular theme among the plumbing posts with a total of 107/360 documents discussing it. Thus, Topic 5 could provide a guide to prioritize future functional requirements implementations. Table 3 below summarizes the results of 10-Fold cross validation of the classifier as well as results obtained for simple keyword matching classification.

To group documents with the same topic, a document can be determined as belonging to a specific topic based on which topic had the highest contribution to that document based on the highest probability scores and assign that topic to that specific document. Table 4 below shows a sample of this step).

Table 4 Sample Document-Topic weights that determine dominant topic of each document

Document	*Topic 0*	*Topic 1*	*Topic 2*	*Topic 3*	*Topic 4*	*Topic 5*	*Topic 6*	*Topic 7*	*Topic 8*	*Topic 9*	*Dominant Topic*
P0.txt	0	0	0	0	0	**0.97**	0	0	0	0	5
P1.txt	0	0	0	0	0	**0.78**	0	0.2	0	0	5
P10.txt	0	0	0	0	0	0	**0.97**	0	0	0	6
P100.txt	0	0	0	0	0	0	0	**0.63**	0	0.34	7
P101.txt	0.01	0.01	0.01	0.01	0.01	0.14	0.01	0.01	0.01	**0.76**	9
P102.txt	0	0	0	0	0	0	0.1	**0.87**	0	0	7

From the results above, the program calculates the topic distribution over the set of documents as shown in the Table 5 below. Clearly, ***Topic 5*** which is represented by words such as "view",

"filter" and "element" is on the top of the list, followed by ***Topic 7*** which is about "Duct" "Level" and "Fitting".

Table 5 Top three recurring words per topic index

	Word 1	*Word 2*	*Word 3*
Topic 0	Parameter	Group	Assembly
Topic 1	Placing	Incorporate	Junction
Topic 2	File	Link	Export
Topic 3	Family	Option	Tag
Topic 4	Calculation	Design	Client
Topic 5	View	Filter	Element
Topic 6	Pipe	Connector	Line
Topic 7	Duct	Level	Fitting
Topic 8	Selection	Object	Command
Topic 9	Schedule	Type	Sheet

Table 6 Topic Distribution across (plumbing) documents

Topic Number	*Number of Documents*
5	107
7	104
9	53
2	29
0	23
3	17
6	16
4	7
8	3
1	1

Table 6 shows the topic distribution across the plumbing posts, which show the total number of posts belonging to a specific topic. This is especially useful in creating the pie chart representation of topic distribution among the data in our prototype dashboard.

4.2.3 PERCEIVED USEFULNESS RESULTS

The perceived usefulness of the prototype is assessed in an attempt to answer RQ6 thoroughly. Perceived usefulness is defined as "the degree to which a person believes that using a particular system would enhance his or her job performance." (Davis, 1989). As previously explained, there were six experts who volunteered to participate in this experiment, of which were five product managers and one support engineer. Semi structured interviews were conducted in a single day with a single participant at a time. A voice recorder was used to capture the interview and any possible insights and opinions expressed by the interviewees. Interviewees were given an

elaborate explanation of the evaluation experiments as well as the tools to be used during the interview.

To avoid bias, the order of using which tool first was alternated between participants. That is, the first participant used the forum first then used the prototype, and the second participant used the prototype first then the forum, and so on. In each part of the experiment, and upon completion of tasks, the participants filled out the perceived usefulness survey. The word usefulness was not mentioned in the experiment nor in the surveys to avoid construct validity threat. The survey was adapted to suite each case; the term prototype was used in the questions when filling the prototype survey, and the term forum was used in the questions similarly. Below is a list of questions used in the perceived usefulness survey adapted from Davis (1989):

*Q1. My job would be difficult to perform without this **prototype**.*
Q2. Using this prototype gives me greater control over my work.
Q3. Using this prototype improves my job performance.
Q4. This prototype addresses my job-related needs.
Q5. Using this prototype saves me time.
Q6. This prototype enables me to accomplish tasks more quickly.
Q7. This prototype supports critical aspects of my job.
Q8. Using this prototype allows me to accomplish more work than would otherwise be possible.
Q9. Using this prototype reduces the time I spend on unproductive activities.
Q10. Using this prototype enhances my effectiveness on the job
Q11. Using this prototype improves the quality of the work I do.
Q12. Using this prototype increases my productivity.
Q13. Using this prototype makes it easier to do my job
Q14. Overall, I find this prototype useful in my job.

Participants were given two tasks to complete within 30 minutes. Participants were asked to perform the tasks using the Autodesk forum and the dashboard prototype and record their findings on paper. The tasks are the following:

Task 1: You are asked to collect top voted 5 functional/requirements or ideas from users in USA. Please write them down in the given paper and or express your findings out loud.

Task 2: While you are asked to seek new opportunities to sell your company's software, you try to make a choice on where to go next to acquire new customers based on the most popular ideas. Write down a list of the top 3 countries you chose or express your findings out loud.

First the participants were interviewed to perform few tasks using the regular Autodesk forum. Second, they were instructed to use the prototype instead to perform the same tasks. After each experiment they were asked to give their feedback on each tool through filling in an adapted version of the perceived usefulness survey by Davis (1989).

***Results**:* This part of the evaluation will present the results of the evaluation surveys used in the experiments.

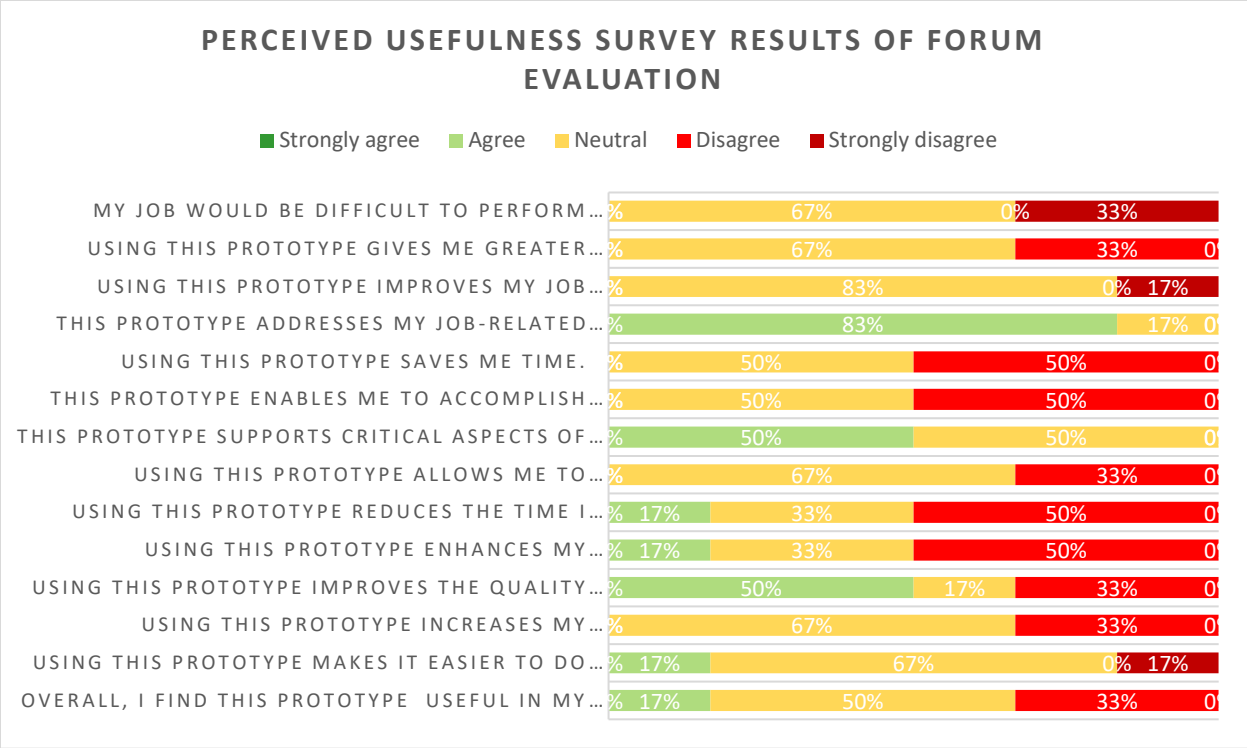

Fig. 15 Forum evaluation survey results

In the first part of the experiment, for evaluating the user forum perceived usefulness, the majority of participants considered it relevant to an extent but not very useful and easy to use. Fig. 15 below illustrates the survey responses of the participants and their answers to the 14 usefulness questions. During the experiment, participants expressed their frustrations when using the website as they had to click on the user profiles just to view the location of the user. This process consumed more time than they expected.

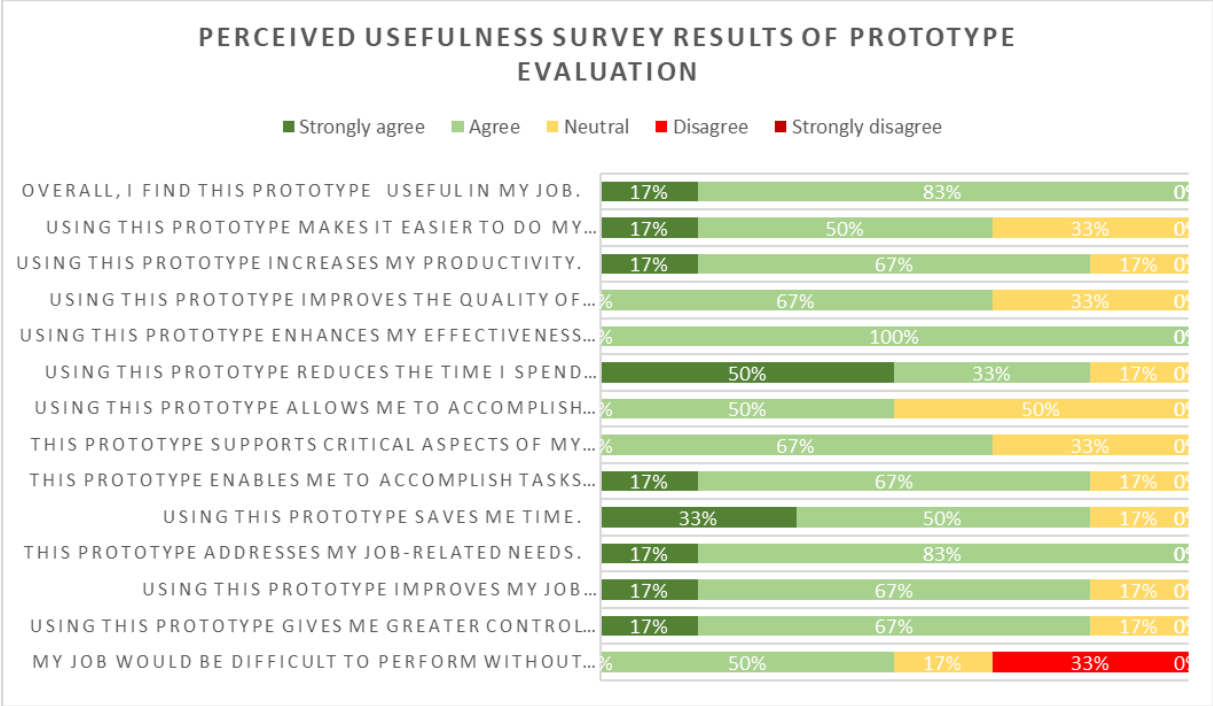

Fig. 16 Results of the perceived usefulness survey for the Dashboard prototype

In the second part of the experiment, for evaluating the dashboard prototype perceived usefulness, the product managers considered it relevant to their daily tasks, very useful and easy to use tool. Fig. 16 summarizes the feedback on the perceived usefulness of the prototype gathered from the participants. What is noteworthy about these results is that the experts agree with at least 50% of the 14 statements of the survey. From the product managers interviews, most comments were about how this prototype can enable them to manage large amount of user feedback, filter highly requested improvements, and discover the appropriate markets for their applications in less time than performing such task manually. A product manager that participated in evaluating the prototype confirmed the impact the prototype by the following quote:

"Nice dashboard which can help with creating a roadmap."

However, there were also some constructive critique provided about the prototype when the participants were asked about possible obstacles for implementing or shortcomings of the prototype. For instance, one product manager said the integration of this prototype to the daily product management task might be a bit complicated as each manager uses different sources of information (and possibly a CRM tool), and thus might take a while for this to change. Two product managers found it unclear and difficult to use the topics filtering functionality. Another manager suggested validating the results with their existing user database and other user fora before the

outputs could be used for any decision making. Additionally, the support engineer wished to have a conclusion or quick report to be added to the prototype. And finally, one manager had no problems using it.

5. CONCLUSION

In today's rapid-growth digital economy, businesses are increasingly dependent on information. In order to adapt to this growth, and gain competitive advantage, businesses seek new innovative approaches and communication channels to extract new software requirements from online user data. Previous and current research publications mainly focus their efforts on exploring social media and mobile application platforms for requirements discovery and extraction. In this research, the focus was on exploring requirement extraction for desktop applications in a highly technical user forum. In this section, we present our answers to the research questions 1 to 6:

For *RQ1: What are the existing studies done on eliciting requirements from user generated content on the Web?*

A literature study was conducted in Chapter 2 and presented existing studies and data-driven approaches that focused on requirements elicitation and extraction from user generated contents and their main findings. Little research studied requirements elicitation from user fora, while social media content and mobile applications platforms were the focus of previous studies.

To answer *RQ2: How to analyze professional user forum posts to identify candidate requirements for software products development?*

From the results in section4.2.1, the Naïve bays classifier was able to achieve an accuracy range between 62% - 90%. This data can be analyzed and visualized by a dashboard as suggested in this research which in turn can provide insights and valuable information regarding highly requested functional requirements and their respective location. Such a dashboard can guide product managers to improve their product and acquire more customers by providing solutions to appropriate markets.

To answer *RQ3: How to analyze Twitter to identify candidate requirements for software products development?*

As discussed in section 2.4, upon human inspection of Twitter content, there was little information that can be used in discovering requirements or any possible complaints or dissatisfaction of users with desktop software Revit. The tweets found contained information such as invitations to BIM related events or conferences or new software version releases. This could be because professional users of desktop software have little use for Twitter in the matter of seeking help, asking for change, reporting issues or requesting functional improvements. Instead, they lean more toward using professional technical communities to get help or voice their needs.

For *RQ4: What is the accuracy of the techniques developed to answer RQ2, RQ3?*

We estimated the average accuracy of this classifier to be **89%**. To further corroborate these results, a fresh sample (on which the classifier has not been trained nor tested) of 45 pre-labeled posts was used to validate the classification model. The model classified 28 correct labels out of the 45, thus, an average of **62% accuracy**. Given the small data set this research had, the accuracy result is considered acceptable especially when compared to the comparable work on mobile app reviews of Maalej and Nabil (2015). Table shows the results of the Naïve Bayes classifier.

For *RQ5: How to use information from Twitter and professional user forums to discover potential new markets?*

We can say based on the prototype in section 3.3 , that it provides useful ways to analyze data from professional user fora and generates visual output, such as the map view of the posts, which enables product managers to immediately discover suitable markets for their next product releases and make actionable decisions accordingly.

Finally, to answer *RQ6: To what extent are the outputs of RQ2, RQ3 and RQ4 perceived as useful by software product managers?*

We found encouraging results from our extensive experiments and, which suggest the usefulness of our proposed prototype in discovering requirement from user forums. It is also found by practitioners to be able to support desktop application providers, product managers and software developers to analyze large amount of user feedback, filter highly requested improvements, and discover the appropriate markets for their applications. There are, however, some drawbacks of the prototype such as the long learning curve of Tableau software as most of the managers have not used it before. Furthermore, the data needs constant update over time to include the most recent and relevant posts.

6. LIMITATIONS AND FUTURE WORK

This section discussed possible threats to the validity of this research and some recommendations for future endeavors to enhance this project.

6.1 Threats to Validity

Internal Validity: There are few threats to the validity of this research. First, there is a possibility of human error in the process of manually classifying posts as bug or feature request. Although we tried to alleviate this threat by following a detailed classification protocol, the possibility of making mistakes cannot be excluded. This could have introduced bias in classifying the data collected. Second, data redundancy is also a threat to validity as we found many users repeatedly post identical ideas or posts to get maximum attention and votes form the community and the Revit forum as well. After much scrutiny in the data, it turned out that some users have written the same posts under different titles in each of the three forum sections (mechanical, electrical and plumbing) in what appears to be an attempt to get a quick response from the software developers. This was not discovered until the final stages of the experiment and thus, could not be prevented. This redundancy in the data could have an influence on the performance of the classifier, and hence the high accuracy results.

External Validity: There could be two factors that may or may not have affected our results. First, it could be that our data is relatively small compared to other studies that used over millions of records from different applications platforms. Second, there might be a threat relates to the generality of our prototype. We validate our prototype with one type of user forum and one company. It is unclear that if our prototype can attain similar results when being applied to other user-focused outlets and other companies.

Construct Validity: During the experiment, it is possible that the subjects' responses and behavior may impose a threat to construct validity as they might respond differently in an attempt to perform better to please the experimenter which can affect the outcome of the experiment (*Evaluation apprehension*).

6.2 Future Work

There are possible steps to enhance this research and further advance the requirements engineering discoveries. First, the context of the research could be expanded to include other type of user forums as well as a collection of user forums.

Second, the size of the case study should be enlarged to include multiple software companies and thus more participants to be able to generalize the results of the prototype usefulness.

Third, prototype technicalities could be further enhanced. The NB classifier should include a penalty to prevent it from biasing to one class over the other and, hence, mistakenly classifying data. The prototype also can benefit from live daily scraping of data and on-spot processing and analysis of information. Additionally, it can be programmed to generate a periodic report on hot topics or trends for instance.

Finally, sources of data could be further explored to include customer-support communication, version control software, or chat records which could manifest a rich source of requirements data.

ACKNOWLEDGEMENT

Special thanks go to NLTK foundation and their book and tutorials were of tremendous help during NLP stages. Special thanks to free python online tutorials authors such as Harrison, Selva Prabhakaran

REFERENCES

Ali, N., & Lai, R. (2016). A method of requirements change management for Global software development. *Information and Software Technology*, *70*, 49–67. https://doi.org/10.1016yj/infsof.2015.09.005

Asur, S., & Huberman, B. A. (2010). Predicting the Future with Social Media. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on* (Vol. 1, pp. 492–499). https://doi.org/10.1109/wi-iat.2010.63

Atefeh, F., & Khreich, W. (2015). A survey of techniques for event detection in Twitter. *Computational Intelligence*, *31*(1), 133–164. https://doi.org/10.1111/coin.12017

Azhar, S., Khalfan, M., & Maqsood, T. (2012). Building information modeling (BIM): now and beyond. *Australasian Journal of Construction Economics and Building*, *12*(4), 15–28. Retrieved from https://search.informit.com.au/documentSummary;dn=013120167780649;res=IELBUS

Baldwin, T., Martinez, D., & Penman, R. B. (2007). Automatic thread classification for Linux user forum information access. In *Proceedings of the Twelfth Australasian Document Computing Symposium (ADCS 2007)* (pp. 72–79).

Bilgram, V., Brem, A., & Voigt, K.-I. (2008). USER-CENTRIC INNOVATIONS IN NEW PRODUCT DEVELOPMENT — SYSTEMATIC IDENTIFICATION OF LEAD USERS HARNESSING INTERACTIVE AND COLLABORATIVE ONLINE-TOOLS. International Journal of Innovation Management, 12(3), 419–458. https://doi.org/10.1142/S1363919608002096

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. Journal of machine Learning research, 3(Jan), 993-1022.

Bird, Steven, Ewan Klein, and Edward Loper (2009), Natural Language. Processing with Python, O'Reilly Media.

Bougie, G., Starke, J., Storey, M.-A., & German, D. M. (2011). Towards understanding twitter use in software engineering: Preliminary findings, ongoing challenges and future questions. *2nd International Workshop on Web 2.0 for Software Engineering, Web2SE 2011, Co-Located with ICSE 2011*, 31–36. https://doi.org/10.1145/1984701.1984707

Brill, O., & Knauss, E. (2011). Structured and unobtrusive observation of anonymous users and their context for requirements elicitation. In *Proceedings of the 2011 IEEE 19th International Requirements Engineering Conference, RE 2011* (pp. 175–184). https://doi.org/10.1109/RE.2011.6051660

Bruns, A., & Burgess, J. E. (2011). The use of Twitter hashtags in the formation of ad hoc publics. In Proceedings of the 6th European Consortium for Political Research (ECPR) General Conference 2011.

Bruns, A., & Stieglitz, S. (2013). Towards more systematic Twitter analysis: Metrics for tweeting activities. International Journal of Social Research Methodology, 16(2), 91–108.

https://doi.org/10.1080/13645579.2012.756095

Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., ... & Layton, R. (2013). API design for machine learning software: experiences from the scikit-learn project. arXiv preprint arXiv:1309.0238.

Carreno, L. V. G., & Winbladh, K. (2013). Analysis of user comments: An approach for software requirements evolution. In *Proceedings - International Conference on Software Engineering* (pp. 582–591). https://doi.org/10.1109/ICSE.2013.6606604

Castro-Herrera, C., Duan, C., Cleland-Huang, J., & Mobasher, B. (2008). Using data mining and recommender systems to facilitate large-scale, open, and inclusive requirements elicitation processes. In *Proceedings of the 16th IEEE International Requirements Engineering Conference, RE'08* (pp. 165–168). https://doi.org/10.1109/RE.2008.47

Chen, N., Lin, J., Hoi, S. C. H., Xiao, X., & Zhang, B. (2014). AR-miner: mining informative reviews for developers from mobile app marketplace. In *Proceedings of the 36th International Conference on Software Engineering - ICSE 2014* (pp. 767–778). https://doi.org/10.1145/2568225.2568263

Cheng, B. H. C., & Atlee, J. M. (2007). Research Directions in Requirements Engineering. In *Future of Software Engineering (FOSE '07)* (pp. 285–303). https://doi.org/10.1109/FOSE.2007.17

Christensen, C. (2011). Twitter revolutions? addressing social media and dissent. *Communication Review*, *14*(3), 155–157. https://doi.org/10.1080/10714421.2011.597235

Coughlan, J., & Macredie, R. D. (2002). Effective communication in requirements elicitation: A comparison of methodologies. *Requirements Engineering*, *7*(2), 47–60. https://doi.org/10.1007/s007660200004

Davis, A., Dieste, O., Hickey, A., Juristo, N., & Moreno, A. M. (2006). Effectiveness of requirements elicitation techniques: Empirical results derived from a systematic review. In *Proceedings of the IEEE International Conference on Requirements Engineering* (pp. 176–185). https://doi.org/10.1109/RE.2006.17

Davis, F. (1989). Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. MIS Quarterly,13(3), 319-340. doi:10.2307/249008

Guzman, E., Alkadhi, R., & Seyff, N. (2017). An exploratory study of Twitter messages about software applications. *Requirements Engineering*, *22*(3), 387–412. https://doi.org/10.1007/s00766-017-0274-x

Guzman, E., & Maalej, W. (2014). How do users like this feature? A fine-grained sentiment analysis of App reviews. In *2014 IEEE 22nd International Requirements Engineering Conference, RE 2014 - Proceedings* (pp. 153–162). https://doi.org/10.1109/RE.2014.6912257

Holstroem, H. (2001). Virtual communities as platforms for product development: an interpretive case study of customer involvement in online game development. *ICIS 2001 Proceedings*, 34. Retrieved from http://aisel.aisnet.org/icis2001/34/?utm_source=aisel.aisnet.org%2Ficis2001%2F34&utm_

medium=PDF&utm_campaign=PDFCoverPages

Hosseini, M., Phalp, K., Taylor, J., & Ali, R. (2014). Towards crowdsourcing for requirements engineering. In *CEUR Workshop Proceedings* (Vol. 1138, pp. 82–87).

Hughes, A. L., & Palen, L. (2009). Twitter adoption and use in mass convergence and emergency events. *International Journal of Emergency Management*, *6*(3/4), 248. https://doi.org/10.1504/IJEM.2009.031564

Java, A., Song, X., Finin, T., & Tseng, B. (2007). Why We Twitter: Understanding Microblogging Usage and Communities. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis - WebKDD/SNA-KDD '07* (pp. 56–65). https://doi.org/10.1145/1348549.1348556

Karlsen, I. K., Maiden, N., & Kerne, A. (2009). Inventing requirements with creativity support tools. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 5512 LNCS, pp. 162–174). https://doi.org/10.1007/978-3-642-02050-6_14

Knauss, A., Borici, A., Knauss, E., & Damian, D. (2012). Towards understanding requirements engineering in IT ecosystems. In *2012 2nd IEEE International Workshop on Empirical Requirements Engineering, EmpiRE 2012 - Proceedings* (pp. 33–36). https://doi.org/10.1109/EmpiRE.2012.6347679

Laurent, P., & Cleland-Huang, J. (2009). Lessons Learned from Open Source Projects for Facilitating Online Requirements Processes. In M. Glinz & P. Heymans (Eds.), *Requirements Engineering: Foundation for Software Quality* (pp. 240–255). Berlin, Heidelberg: Springer Berlin Heidelberg.

Lim, S. L., & Finkelstein, A. (2012). StakeRare: Using social networks and collaborative filtering for large-scale requirements elicitation. *IEEE Transactions on Software Engineering*, *38*(3), 707–735. https://doi.org/10.1109/TSE.2011.36

Maalej, W., & Nabil, H. (2015). Bug report, feature request, or simply praise? On automatically classifying app reviews. In *2015 IEEE 23rd International Requirements Engineering Conference, RE 2015 - Proceedings* (pp. 116–125). https://doi.org/10.1109/RE.2015.7320414

Maalej, W., Nayebi, M., Johann, T., & Ruhe, G. (2016). Toward data-driven requirements engineering. *IEEE Software*, *33*(1), 48–54. https://doi.org/10.1109/MS.2015.153

Magnusson, P. R., Matthing, J., & Kristensson, P. (2003). Managing User Involvement in Service Innovation: Experiments With Innovating End Users. *Journal of Service Research*, *6*(2), 111–124. https://doi.org/10.1177/1094670503257028

Martin, W., Sarro, F., Jia, Y., Zhang, Y., & Harman, M. (2017). A Survey of App Store Analysis for Software Engineering. *IEEE Transactions on Software Engineering*, *43*(9), 817–847. https://doi.org/10.1109/TSE.2016.2630689

Mendoza, M., Poblete, B., & Castillo, C. (2010). Twitter under crisis. In *Proceedings of the First Workshop on Social Media Analytics - SOMA '10* (pp. 71–79). https://doi.org/10.1145/1964858.1964869

Mich, L., Anesi, C., & Berry, D. M. (2005). Applying a pragmatics-based creativity-fostering technique to requirements elicitation. *Requirements Engineering*, *10*(4), 262–275. https://doi.org/10.1007/s00766-005-0008-3

Morales-Ramirez, I., Kifetew, F. M., & Perini, A. (2017). Analysis of online discussions in support of requirements discovery. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 10253 LNCS, pp. 159–174). https://doi.org/10.1007/978-3-319-59536-8_11

Nambisan, S. (2002). Designing virtual customer environments for new product development: Toward a theory. *The Academy of Management Review*, *27*(3), 392–413.

Nayebi, M., Cho, H., Farrahi, H., & Ruhe, G. (2017). App store mining is not enough. In *Proceedings - 2017 IEEE/ACM 39th International Conference on Software Engineering Companion, ICSE-C 2017* (pp. 152–154). https://doi.org/10.1109/ICSE-C.2017.77

Nayebi, M., Quapp, R., Ruhe, G., Marbouti, M., & Maurer, F. (2017). Crowdsourced exploration of mobile app features: A case study of the fort mcmurray wildfire. In *Proceedings - 2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering in Society Track, ICSE-SEIS 2017* (pp. 57–66). https://doi.org/10.1109/ICSE-SEIS.2017.8

Nuseibeh, B., & Easterbrook, S. (2000). Requirements engineering: a roadmap. *Proceedings of the Conference on The Future of Software Engineering - ICSE '00*, *1*, 35–46. https://doi.org/10.1145/336512.336523

Pagano, D., & Maalej, W. (2013). User feedback in the app store: An empirical study. In *2013 21st IEEE International Requirements Engineering Conference, RE 2013 - Proceedings* (pp. 125–134). https://doi.org/10.1109/RE.2013.6636712

Reddivari, S., Rad, S., Bhowmik, T., Cain, N., & Niu, N. (2014). Visual requirements analytics: a framework and case study. Requirements engineering, 19(3), 257-279.

Romero, D., & Molina, A. (2011). Collaborative networked organisations and customer communities: Value co-creation and co-innovation in the networking era. In *Production Planning and Control* (Vol. 22, pp. 447–472). https://doi.org/10.1080/09537287.2010.536619

Rouse, A. (2010). A Preliminary Taxonomy of Crowdsourcing. *ACIS 2010 Proceedings*, 76. https://doi.org/10.1145/1400214.1400244

Ruhe, G., Nayebi, M., & Ebert, C. (2017). The Vision: Requirements Engineering in Society. In *2017 IEEE 25th International Requirements Engineering Conference (RE)* (pp. 478–479). https://doi.org/10.1109/RE.2017.70

Sandor, A., Lagos, N., Vo, N.-P.-A., & Brun, C. (2016). Identifying User Issues and Request Types in Forum Question Posts Based on Discourse Analysis. In *Proceedings of the 25th International Conference Companion on World Wide Web* (pp. 685–691). Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee. https://doi.org/10.1145/2872518.2890568

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

Services and Applications, *6*(1). https://doi.org/10.1186/s13174-015-0021-9

Sherief, N., Abdelmoez, W., Phalp, K., & Ali, R. (2015). Modelling users feedback in crowd-based requirements engineering: An empirical study. In *Lecture Notes in Business Information Processing* (Vol. 235, pp. 174–190). https://doi.org/10.1007/978-3-319-25897-3_12

Snijders, R., Ozum, A., Brinkkemper, S., & Dalpiaz, F. (2015). Crowd-Centric Requirements Engineering: A method based on crowdsourcing and gamification. *Department of Information and Computing Sciences, Utrecht University, Tech. Rep. UU-CS-2015-004*, (March).

Sondhi, P., Gupta, M., Zhai, C., & Hockenmaier, J. (2010). Shallow information extraction from medical forum data. *Proceedings of the 23rd International Conference on Computational Linguistics: Posters (COLING '10)*, (August), 1158–1166. Retrieved from http://dl.acm.org/citation.cfm?id=1944566.1944699

Stieglitz, S., Brockmann, T., & Dang-Xuan, L. (2012). Usage of Social Media for Political Communication. *16th Pacific Asia Conference on Information Systems*, 22.

Cook, K. A., & Thomas, J. J. (2005). Illuminating the path: The research and development agenda for visual analytics (No. PNNL-SA-45230). Pacific Northwest National Lab. (PNNL), Richland, WA (United States).

Tumasjan, A., Sprenger, T. O., Sandner, P. G., & Welpe, I. M. (2010). Predicting Elections with Twitter. In *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media* (pp. 178–185). Retrieved from http://www.aaai.org/ocs/index.php/ICWSM/ICWSM10/paper/viewFile/1441/1852

Wang, L., Kim, S. N., & Baldwin, T. (2012). The Utility of Discourse Structure in Identifying Resolved Threads in Technical User Forums. In *COLING* (pp. 2739–2756). Mumbai.

Wanner, F., Ramm, T., & Keim, D. a. (2011). ForAVis: explorative user forum analysis. *Proceedings of the International Conference on Web Intelligence, Mining and Semantics (WIMS)*, 14. https://doi.org/10.1145/1988688.1988705

Weimer, M., Gurevych, I., & Mühlhäuser, M. (2007). Automatically assessing the post quality in online discussions on software. *Proceedings of the ACL*, (June), 125–128. https://doi.org/10.3115/1557769.1557806

APPENDICES

1 Appendix A: Classification Data

Fig. 17 Cronbach's Alpha for inter-coder agreement

Reliability Statistics

Cronbach's Alpha	N of Items
,831	3

Protocol for Manual Classification

For tagging a file as **Feature request:** In order to tag a file as Feature request it has to adhere to the following rules:

1. Contain any of the keywords(or similar meaning): {add, please, could, would, hope, implement, improve, miss, request, should, suggest, want, wish, feature, support, include, integrate, to be able, possible, allow, complaint, if only, improvement, instead of, lacks, look forward to, maybe, missing, must, needs, please, prefer, waiting for, will, beneficial, handy, good idea
2. Asks explicitly for new functionality that apparently does not exist in the current software
3. Or has sentences like “nice to have this”, “possibility to” etc.

For tagging **Bug report:** In order to tag a file as Bug report it has to adhere to the following rules:

1. It includes any of the terms (or similar meaning) {bug, fix, problem, issue, not working, stuck, freeze, not responding, defect, unable, crash, solve, couldn’t, can’t, seem to, won’t, do not, crappy}
2. It explicitly describes a problem or something that is not working properly
3. Might contain fix suggestion

EXAMPLE	**BUG**	**FEATURE**
Why Revit do not have "Control+A" !!! it is a simple wish,,, to be able to select everything in your model without "Window selection" ,,,,as you may miss elements hides anywhere ,,,,Select All ,,,,Control A ,,,is as simple as to be implemented in almost all applications "CAD" why not our lovely "Revit"!!		X
For some unknown reason Shafts cannot be tagged. And all workaround methods to tag Shafts suck	X	

For the classification with Naive Bayes algorithm, a trial of different counts of informative words in the corpus that the algorithm needed to build the model on was executed from 100 words up to 2000 words. The respective statistical measures were calculated for each count of informative words. After few tests, 300-word features were selected as a reasonable average word count as it produced more stable statistical measures and meaningful informative word features.

Table 7 Classification accuracies for different word features

		Feature Request Class			Bug Class		
WordFeatures	**Accuracy**	*Precision*	*Recall*	*F-measure*	*Precision*	*Recall*	*F-measure*
100	0.8643	*0.8750*	*0.9680*	*0.9191*	*0.7842*	*0.4578*	*0.5779*
200	0.8997	*0.9061*	*0.9730*	*0.9383*	*0.8652*	*0.6321*	*0.7304*
300	0.8969	*0.9252*	*0.9473*	*0.9361*	*0.7704*	*0.6980*	*0.7324*
400	0.8978	*0.9149*	*0.9598*	*0.9368*	*0.8154*	*0.6653*	*0.7326*
500	0.9055	*0.9355*	*0.9472*	*0.9413*	*0.7773*	*0.7391*	*0.7575*
600	0.9275	*0.9738*	*0.9323*	*0.9526*	*0.7894*	*0.9100*	*0.8454*
700	0.9150	*0.9473*	*0.9460*	*0.9466*	*0.7883*	*0.7931*	*0.7905*
800	0. 9079	*0.9413*	*0.9414*	*0.9414*	*0.7843*	*0.7841*	*0.7841*
900	0.9369	*0.9637*	*0.9585*	*0.9611*	*0.8223*	*0.8431*	*0.8323*
1000	0.9293	*0.9495*	*0.9605*	*0.9549*	*0.8532*	*0.8183*	*0.8353*
2000	0.9429	*0.9837*	*0.9439*	*0.9634*	*0.8111*	*0.9389*	*0.8703*

Table 8 10-fold cross validation for NB classification and simple keyword matching results

Type of classification	***Type/class of post***	***Ave. Precision***	***Ave. Recall***	***Ave. F-measure***
Naive Bayes Classifier	*Bug Report*	0.9736	0.9291	0.9508
	Feature Request	0.9365	0.9762	0.9558
Simple keyword matching	*Bug Report*	0.5	0.25	0.333
	Feature Request	0.505	0.754	0.605

The above table summarizes the results of both the Naïve Bayes classifier and simple keyword matching done to re-validate the accuracy.

Topic Modelling Outputs

Mechanical Topic Modelling:

Below are topic modelling related outputs.

Table 9 Most recurring three words per topic

	Word 1	*Word 2*	*Word 3*
Topic 0	Family	Plan	Section
Topic 1	Parameter	Element	Type
Topic 2	Fabrication	File	Design
Topic 3	Sheet	Keynote	Scale
Topic 4	Phase	Calculation	Equipment
Topic 5	Group	Material	Control
Topic 6	View	Line	Level
Topic 7	Filter	Color	Override
Topic 8	Symbol	Option	Unit
Topic 9	Model	Number	Schedule

Table 10 Topics distribution across mechanical posts

Topic Number	*Number of Documents*
1	201
6	105
0	72
2	50
9	27
8	22
3	17
5	9
7	5
4	4

Electrical Topic Modelling:

Table 11 Top recurring three words per topic index

	Word 1	*Word 2*	*Word 3*
Topic 0	Panel	Schedule	Circuit
Topic 1	Filter	Material	View
Topic 2	Bend	Message	Radius
Topic 3	c4r	Adding	Central
Topic 4	File	ifc	Save
Topic 5	Tag	Sheet	Space
Topic 6	View	Family	Conduit
Topic 7	Line	Text	Phase
Topic 8	Switch	Light	Filter
Topic 9	Tray	Cable	Parameter

Table 12 Topics distribution across electrical posts

Topic Number	*Number of Documents*
6	222
0	120
7	47
9	32
5	31
1	31
8	15
4	5
3	1
2	1

2 Appendix B: Qualitative Evaluation

Experiment Protocol

You will participate in evaluating a dashboard prototype as a part of the MSc thesis project of Amasi Elbakush, Utrecht University. By participating in this experiment, you are agreeing to share your opinion with the audience of this research.

This prototype aims at helping product managers in finding functional requirements relevant to their product without the need for direct customer contact. Please bear in mind that this is just a prototype and thus some functionalities or views may be rough. **Duration of this experiment: 30 minutes.** You can leave any time you wish.

- **WHAT TO DO**

  You will have to repeat the tasks below twice. First, you need to perform the same tasks using the prototype dashboard, and finally fill in the survey . Second, You will perform the tasks below using Revit Forum website, then fill in this survey.

  Task 1: You are asked to collect top voted 5 functional/requirements or ideas from users in USA. Please write them down in the given paper and or express your findings out loud.

  Task 2: While you are asked to seek new opportunities to sell your company's software, you try to make a choice on where to go next to acquire new customers based on the most popular ideas. Write down a list of the top 3 countries you chose or express your findings out loud.

Thank you!

3 Appendix C: Code Scripts used

Note: All programming was done in Python programming language

Scraping code:

```
import scrapy

from scrapy import Request

from revitForum.items import RevitForumItem

class PlumbingPostsSpider(scrapy.Spider):

    name = 'plumbingposts'

    allowed_domains = ["forums.autodesk.com"]

    start_urls = ['https://forums.autodesk.com/t5/revit-ideas/idb-
p/302/tab/most-recent/label-name/plumbing']



    def parse(self, response):

        jobs = response.xpath('//div[@data-lia-message-uid]')

        for job in jobs:

            item = RevitForumItem()

            item['ID'] = str(job.xpath('.//div/@data-message-
id').extract_first())

            item['Status'] = job.xpath('.//span[@class="lia-message-subject-
status"]/span/a[@class="lia-link-navigation message-status-
link"]/text()').extract()

            item['Date'] = job.xpath('.//span[@class="DateTime lia-message-
posted-on lia-component-common-widget-date"]/span[@class="local-
date"]/text()').extract_first("")[1:]

            item['Title'] = job.xpath('.//a[@class="lia-link-navigation idea-
article-link"]/text()').extract_first()

            description = " ".join(line for line in
job.xpath('.//div[@class="lia-message-body-
content"]/p/text()').extract()).strip(' \t\n\r')

            if not description.strip():
```

```
                description = " ".join(line for line in
job.xpath('.//div[@class="lia-message-body-
content"]/p/span/text()').extract()).strip(' \t\n\r')

            if not description.strip():

                description = " ".join(line for line in
job.xpath('.//div[@class="lia-message-body-
content"]/p/span/p/text()').extract()).strip(' \t\n\r')

            if not description.strip():

                description = " ".join(line for line in
job.xpath('.//div[@class="lia-message-body-
content"]/p/font/text()').extract()).strip(' \t\n\r')

            item['Description'] = description

            item['Vote'] = job.xpath('.//span[@class="MessageKudosCount lia-
component-kudos-widget-message-kudos-count"]/text()').extract_first().strip('
\t\n\r')



            yield item

        relative_next_url = response.xpath('//li[@class="lia-paging-page-next
lia-component-next"]/a/@href').extract_first()

        absolute_next_url = response.urljoin(relative_next_url)


        yield Request(absolute_next_url, callback=self.parse)

#End of Program
```

Classification code:

```
import re

import os

import sys

import nltk

import string

import random

import numpy as np

from nltk.corpus import stopwords

from nltk.stem import WordNetLemmatizer

from nltk.corpus import CategorizedPlaintextCorpusReader

from nltk.classify import ClassifierI

from nltk.tokenize import word_tokenize

from statistics import mode

reload(sys)

sys.setdefaultencoding('Cp1252')

mydir = 'C:/Users/Amasi/revitForum/Classification'

mr= CategorizedPlaintextCorpusReader(mydir, r'.*\.txt',
cat_pattern=r'(\w+)/*', encoding="Latin-1")

documents = [(list(mr.words(fileid)), category)for category in
mr.categories()for fileid in mr.fileids(category)]

random.shuffle(documents)

stop = set(stopwords.words('english'))

string_punctuation = set(string.punctuation)

stop.update(['however','when','revit','autocad','what',

'why','thanks','cant','would','could','nice','Hi','Hello','it','It'])

lemmatizer = nltk.WordNetLemmatizer()

def refine(document):

      refined_doc= []
```

```
      for w in document:
      w = w.lower()
      w = re.sub(r'[^\x00-\x7F]+',' ', w)
      w= w.encode("Latin-1")
      if w not in stop and w not in string_punctuation:
      word = lemmatizer.lemmatize(w)
      refined_doc.append(word)
      return refined_doc
refined_docs = [(list(refine(doc)), category)for (doc, category) in
documents]
all_words = []
for w in mr.words():
      all_words.append(w.lower())
      all_words = nltk.FreqDist(all_words)
      word_features = list(all_words.keys())[:300]
def find_features(document):
      words = set(document)
      features = {}
      for w in word_features:
            features[w] = (w in words)
      return features
featuresets = [(find_features(rev), category) for (rev, category) in
documents]
training_set = featuresets[:100]
classifier = nltk.NaiveBayesClassifier.train(training_set)
directory = "C:/Users/Amasi/revitForum/untested/"
for root, dirs, files in os.walk(directory):
for f in files:
      print(f)
```

```
      filename = os.path.join(directory, f)
with open(filename, 'r') as fin:
for line in fin:
      doc_tokens = word_tokenize(line.lower())
      doc = ' '.join(map(lambda s: re.sub(r'[^\x00-\x7F]+',"", s), doc))#to
remove any non-ascii chars
featurized_doc ={i:(i in doc) for i in word_features}
tagged_label = classifier.classify(featurized_doc)
print(tagged_label)
# END of Program
```

Topic Modelling code:

```
from sklearn.feature_extraction.text import TfidfVectorizer,
CountVectorizer

from sklearn.decomposition import NMF, LatentDirichletAllocation

import numpy as np

import string

import os

import warnings

import nltk

import random

from nltk.corpus import stopwords

from nltk.stem import WordNetLemmatizer

from nltk.classify.scikitlearn import SklearnClassifier

import collections

import numpy as np

import gensim

from gensim import corpora

from autocorrect import spell

from gensim import corpora

import pyLDAvis.gensim

from nltk.corpus.reader.plaintext import PlaintextCorpusReader


warnings.filterwarnings('ignore')

stop = set(stopwords.words('english'))

extra =
['revit','autodesk','however','when','what','why','this','thanks','are','woul
d','could','nice','hi','hello','it','i','us','u', 'able','a']

exclude = set(string.punctuation)

lemma = WordNetLemmatizer()
```

```
      def cleanDoc(doc):
          if not doc:
              return ''
          else:
              stop_free = " ".join([i for i in doc.lower().split() if i not
in stop])
              punc_free = ''.join(ch for ch in stop_free if ch not in
exclude)
              extra_free = " ".join([d for d in punc_free.lower().split() if
d not in extra])
              normalized = " ".join(lemma.lemmatize(w) for w in
extra_free.split())
          return normalized


      def display_topics_cluster_docs(H, W, feature_names, documents,
no_top_words, no_top_documents, cleandictionary, dirname):
          for topic_idx, topic in enumerate(H):
              print "Topic %d:" % (topic_idx)
              topwords = "-".join([feature_names[i]
                                   for i in topic.argsort()[:-no_top_words - 1:-
1]])
              print "-".join([feature_names[i]
                                   for i in topic.argsort()[:-no_top_words - 1:-
1]])
              top_doc_indices = np.argsort( W[:,topic_idx] )[::-
1][0:no_top_documents]
              for doc_index in top_doc_indices:
                  for k,v in cleandictionary.iteritems():
                      if v == (documents[doc_index]).encode("Latin-1"):
                          dir_name = dirname + "/" + str(topic_idx) + "-" +
str(topwords)
```

```
                            file_name = dir_name + "/" + (str(k).rsplit('/',
1)[-1])
                            if not os.path.exists(dir_name):
                                os.makedirs(dir_name)
                                fp = open(file_name,"w")
                                fp.write((documents[doc_index]).encode("Latin-
1") )
                                fp.close()
                            print("{} ".format(k))
                    print ((documents[doc_index]).encode("Latin-1"))


      mydir = 'C:/Users/Amasi/revitForum/posts_data/'
      corpus = PlaintextCorpusReader(mydir, ".*\.txt", encoding="Latin-1")
      #Accessing the name of the files of the corpus
      files = corpus.fileids()
      cleandictionary ={}
      diction={}
      documents = []
      for filename in files:
          doc_value = corpus.raw(filename)
          if not doc_value:
              doc_value = " "
          diction[filename]= doc_value.encode("Latin-1")
          cleand = cleanDoc(doc_value)
          cleandictionary[filename]= cleand
          documents.append(cleand)


      tfidf_vectorizer = TfidfVectorizer(max_df=0.95, min_df=2,
stop_words='english')
      tfidf = tfidf_vectorizer.fit_transform(documents)
```

```
      tfidf_feature_names = tfidf_vectorizer.get_feature_names()


      tf_vectorizer = CountVectorizer(max_df=0.95, min_df=2,
stop_words='english')

      tf = tf_vectorizer.fit_transform(documents)

      tf_feature_names = tf_vectorizer.get_feature_names()

      no_topics = 10

      lda_model = LatentDirichletAllocation(n_topics=no_topics, max_iter=5,
learning_method='online', learning_offset=50.,random_state=0).fit(tf)

      lda_W = lda_model.transform(tf)

      lda_H = lda_model.components_

      no_top_words = 3

      no_top_documents = 50

      print("----* Topics using LDA algorithm, followed by related posts: *--
--")

      print("-" * 20)

      print("\n")

      display_topics_cluster_docs(lda_H, lda_W, tf_feature_names, documents,
no_top_words, no_top_documents, cleandictionary, "topic_modelling-LDA")

      print("\n END OF PROGRAM")
```