Acta Scientiarum

http://periodicos.uem.br/ojs
ISSN on-line: 1807-8664
Doi: 10.4025/actascitechnol.v44i1.58658

COMPUTER SCIENCE

Extracting feature requests from online reviews of travel industry

Superna Kumari and Zulfiqar Ali Memon*

Department of Computer Science, National University of Computer and Emerging Sciences, ST-4, Sector 17-D, Shah Latif Town (on National Highway), Karachi, Sindh, Pakistan. *Author for correspondence. E-mail: memon.zulfiqar@gmail.com

ABSTRACT. Before product development, Requirement Engineering (RE) is the fundamental need to know customer preferences for any product. Traditionally, RE is carried out in several ways, particularly by conducting interviews, questionnaires, surveys etc. but these methods provide limited amount of data. As user's preferences vary from country to country for any type of application, it is very hectic and time consuming to collect user requirements from different countries manually. As the internet is widely used now a days, a large number of customer's reviews are available online that can be used to obtain the requirements for any product without manual work. Online customer reviews can be divided into three types: user experience, bugs and feature requests. Among these 3 categories, feature requests can be very useful for stakeholders (analysts/ requirements engineers) to acquire the requirements of each application. So, the approach is proposed for feature requests extraction from mobile application reviews of travel industry. In this paper, 4 categories of mobile apps of travel industry belonging to 5 countries have been extracted from Google Play Store and Apple Store. For each category, data from 5 different mobile applications have been considered. Since, the review of users from different countries is in their respective language, those reviews are translated into a standard language i.e. English, and then feature requests from these reviews have been extracted. After that, features are retrieved from user reviews and topic modeling is performed on extracted features since one or more features can be modelled under one topic. To know the opinions of users for any feature request, sentiment analysis is also performed on feature request sentences. These feature requests are also classified as Functional and Non-functional Requirements since it is very useful for application developers to improve or maintain the product to better facilitate the application users.

Keywords: Requirement engineering; feature requests; user reviews; app stores; travel booking applications; sentiment analysis

Received on April 14, 2021.
Accepted on June 6, 2021.

Introduction

Requirement gathering is the basic necessity of finding the customer's preferences for any application. Requirement engineering (RE) helps stakeholders and the organization to get customer preferences and meet customer needs. (Ghanyani, Murad, & Mehmood, 2018). Another way of RE is crowdsourcing but it has been observed that many users do not participate in crowdsourcing or the results can be unrealistic or may not be from right users of the application. (Besrour, Rahim, & Dominic, 2016).

Traditionally, gathering the requirements for each application requires manual work. There are 12 RE techniques that include techniques such as conducting interviews, surveys, questionnaires, brainstorming, peer reviews, etc. (Besrour et al., 2016). However, through these techniques, limited data is obtained from a limited number of customers. Traditional RE methods require reaching target customers, which can be very hectic and time-consuming, especially when customers are from different parts of the world and enough customers need to be reached to collect a good amount of customer reviews in order to analyze customer needs.

Nowadays, a large number of customers are available online all over the world and their reviews for software product are available on different platforms and channels like social media sites, mobile app stores, discussion forums, etc. As the use of mobile apps increases day by day, many software are available in form of mobile apps. Among the mentioned platforms, App Stores are the most used platform that helps users find, purchase (if the software is paid), and install mobile apps. Users can also submit their feedback in form of star ratings or text reviews on the App Stores. There are many app stores such as Google Play Store, Apple Store, Samsung App Store, Blackberry App Store, but Google Play Store and Apple Store are widely used around the world. The ratings on these platforms are in the form of star ratings which shows the average rating given to this app by app users. These ratings can show the usefulness of the application in the form of a number usually from 1 to 5 stars. However, the ratings do not show the details of the customer reviews and their in-depth opinions. Text reviews provided by users of the application may contain detailed opinions of users. These reviews can be divided into 3 categories: user experience, bugs and feature suggestions / feature requests. User Experience shows the stories and experiences they've had using the app. This can give stakeholders an idea of how the app has been used. It can also provide new ideas of use for which the app was not originally developed and can provide additional functionality related to it, bugs are the problems that users encounter while using application and feature requests may include suggestions for new features or improvements of existing features. These reviews are useful not only for users who want to know the opinion of other users of the application, but also for stakeholders (requirements analysts / engineers) to capture customer requirements in order to improve their application. Many software companies collect implicit feedback from usage data, error logs, sensor data, or via traditional methods, but reviews from these app stores show explicit feedback from software users which they submit after using the app. Feature requirements available in user reviews can be new, interesting, and useful for users of applications that developers may not think of for themselves. Since mobile apps are provided to facilitate different parts of the world, a large number of mobile app user reviews are available online from all over the world. Since it is easy to collect customer reviews from different parts of the world, but extracting feature requests for the product for each country by manually analyzing customer reviews is a very hectic, tedious and manual work and requires human intelligence.

In order to facilitate requirement engineers and analysts to determine the most informative reviews and retrieve feature requests / suggestions for any application based on the country in which the application is used, an approach is proposed to extract new feature requests from country-based online customer reviews for various categories of mobile applications of travel industry. This approach involves the technique that uses various Natural Language Processing (NLP) and Machine Learning (ML) techniques to extract feature requirements or suggestions from user reviews. It collects user reviews of different categories of mobile apps of the travel industry based on country from Google Play Store and Apple Store. After collecting user reviews, reviews containing feature requests are retrieved. Then features are extracted from those reviews. Since more than one feature can be modelled under one topic, so Topic Modeling is performed. Then to find the opinions of customers from reviews, sentiment analysis is performed then feature requests are classified as Functional or Non-Functional Requirements.

So far, several studies have proposed approaches for feature extraction focusing only the best generalized app store apps. This paper focus on several categories of travel industry mobile applications used in different countries and try to identify the needs of users according to country. Since this paper focuses on the travel industry, categories of Mobile Apps such as car reservations, coach reservations, flight reservations and train reservations applications have been considered. For the implementation of proposed approach, customer review data from 5 mobile applications of each mentioned category belonging to 5 different countries (France, Russia, Korea, Turkey and Spain) have been used.

The rest of this paper is organized as follows. Section II, describes the prior studies related to this work, Section III elaborates proposed methodology, Section IV explains experiment results and evaluation and Section V describes conclusion and suggestion for future work.

Related work

RE is the fundamental and most important part of any software development process. It is very necessary for software product stakeholders to meet customer needs by developing new product or improving their existing product. A lot of work has been done to facilitate the RE process both in a traditional and automated way.

To collect the requirements traditionally, Besrour et al. (2016) carried out study on various traditional RE processes to help stakeholders select the most appropriate RE process for small and medium-sized software companies. Authors examined 15 techniques, 12 of which were supported by software professionals with industry experience. A survey on a 5-point scale was used to measure the importance of RE techniques which is as follows: 1 = strongly disagree, 2 = disagree, 3 = neutral, 4 = agree and 5 = strongly agree. Techniques that have received strong support include interviews, brainstorming, structured NL, JAD, peer review, goal orientation, activity diagram, soft requirements specification, checklist, ERD based specification, user case, Laddering and Misuse case.

Crowdsourcing is another way to collect requirements. Ghanyani et al. (2018) examined the requirements elicitation through crowdsourcing and their comparison with traditional RE concepts shows the advantage of crowdsourcing over traditional methods. In crowdsourcing, the crowd of app users have been involved in the requirements development process and they can also check their own requirements if those are adopted for app functionality. This reduces the costs of outsourced analysts and their workload. It can create unrealistic expectations and getting quality demands from the crowd can also be a difficult task as they are not motivated to participate in the process. This paper suggests motivating users to participate in the requirements elicitation process by offering them incentives. After conducting a survey based on 9 categories that include largeness, anonymity, diversity, competence, intrinsic motivation, collaboration, volunteering, extrinsic incentives, feedback, Authors found that gamification is the only motivation for the crowd to participate.

Mobile applications have been available online now a days, and app stores allow users to search, purchase (if paid) and install mobile apps and then submit feedback in the form of reviews and ratings mentioned by Genc-Nayebi and Abran (2017).

The app stores are successful now a days, as they are used by a large number of customers and their reviews are also available in the app stores. Martin, Sarro, Jia, Zhang, and Harman (2017) carried out several analyzes of the App Stores to allow stakeholders to improve the product from different perspectives. Authors reviewed the App Store based on technical and non-technical attributes which show its analysis from different perspectives such as feature analysis, API analysis, release engineering, customer review analysis and Security.

The usefulness of reviews motivates software stakeholders to analyze customer reviews to improve their application. Charrada (2016) collected and listed 32 factors that influenced the usefulness of customer reviews on products. These factors have been divided into 5 categories: 1) language, 2) volume and longevity, 3) ratings, sentiments and emotions, 4) content and 5) reviewers. Authors examined requirement engineer's tasks such as evaluation and processing to find the applicability of factors in the RE process.

A lot of work has been done to analyze customer feedback. Genc-Nayebi and Abran (2017) worked to identify the solutions currently available for extracting opinions from reviews, challenges and unresolved problems in this domain of the App Store. The ratings and reviews available in the app stores are user-driven reviews that can help improve the quality of the software and fix the missing app functionality. Extracting opinions from App Store reviews has many challenges, including the unstructured nature of user reviews, the colloquial language used, reading large amounts of reviews, spam detection of opinions or fake reviews. While reviewing literature on opinion mining techniques, Authors found that most of the studies were based on manual classification and correlation analysis, the automated extraction of app functionality did not consider the nature of the review text, the information required by developers and users are different. External sources such as app crash reports, tweets, etc. could be used to enrich data analysis.

Maalej, Nayebi, Johann, and Ruhe (2016) discussed how software developers and analysts can use user data available in application stores to identify, prioritize and manage software product requirements. This paper demonstrated that feedback can be collected explicitly through user feedback and implicitly through the collection of usage data, interaction tracks and logs. Nagappan and Shihab (2016) discussed the most recent and future research trends at different stages of the software development life cycle: requirements (including non-functional), design and development, testing and maintenance. They discussed it in terms of requirements, energy, security, development, testing, maintenance, and monetization.

Sohail, Siddiqui, and Ali (2016) presented a method to extract features from online reviews and perform analysis on these features to recommend books using opinion mining techniques. This paper represents extraction of feature reviews using human intelligence and classification of books into 7 categories, namely (1) frequency of occurrence on the search engine results page (SERP), (2) useful content, (3) sufficient material, (4) Extraneous content, (5) market availability, (6) physical attributes, and (7) price.

Malik and Shakshuki (2016) worked on mining collective opinions for the comparison of mobile applications. Authors worked to extract the most talked about features from the online reviews available in the app stores. They proposed a methodology that first performs some preprocessing tasks to filter explicit sentences containing requirements, clean up the data and remove noise from the data, then extract the noun, verb, and adjective via NLTK. Term Frequency – Inverse Document Frequency (TFIDF) is then applied to extract the feature. Then sentiment analysis is performed on reviews in which people's opinions on app functionality is categorized into: positive, negative and natural. WordNet16 was used to divide some sentimental words into six types: disgust, anger, fear, sadness, surprise and joy.

Sentiment analysis is the process of analyzing human feelings on any topic and classify them as positive, negative, or neutral based on user opinions on that topic available in the text. The sentiments of the users can be expressed implicitly or explicitly. Chandankhede, Devle, Waskar, Chopdekar, and Patil (2016) worked on analyzing the implicit sentiments of users' reviews. This paper proposed a methodology consisting of modules such as data gathering/ acquisition, preprocessing, feature extraction, classification. Data acquisition involves acquiring user reviews from various sources such as forums, app stores, etc. Pre-processing involves multiple data cleansing processes then POS tagging is applied on the data. Authors have used five aspects such as food, service, hygiene, atmosphere and price and extracted synonyms for each aspect of the reviews and matched these synonyms in each category and performed sentiments classification on opinion words.

Guzman and Maalej (2014) proposed a technique to help stakeholder to preprocess, cluster and analyze user opinions using NLP techniques. Authors developed a tool that uses Google Play APIs to extract reviews from Google App stores and store this data in the MYSQL database. Then retrieved data is cleaned and features are identified by extracting the nouns, verb and adjective from the cleaned data. The identified feature of the application is then provided as input to the collocation finding algorithm. After extracting the features, the user's sentiments about those features are extracted where the features are assigned a quantitative (i.e. positive or negative) value. Topic modeling is then performed using Latent Dirichlet Allocation (LDA) to cluster the fine-grained features into more meaningful high-level features. Then recall, precision and F-measure are used for result evaluation.

Harman, Jia, and Zhang (2012) proposed the technique to retrieve feature information from app store reviews and combine it with information available to analyze technical (app descriptions), Business (downloads and pricing) and customer aspects of applications (qualitative and quantitative data). This paper proposed an approach that consists of data extraction using a web crawler, parsing using a set of pattern templates, preprocessing using different data cleaning techniques, mining features using word frequency and collocation analysis and data correlation analysis using Spearman's rank correlation technique. The outcome of technique show a strong correlation between customer ratings and app download rankings.

Iacob and Harrison (2013) developed a prototype to extract feature requests from online reviews of mobile applications. The designed system is based on multiple phases that include the extraction of available applications reviews by implementing a web crawler, the splitting of the reviews into sentences, Defining set of linguistic rules to refer to sentences containing requests, feature extraction, defining a set of rules to summarize requests for the retrieved features, evaluating these techniques using the confusion metric.

Phetrungnapha and Senivongse (2019) proposed the method of classifying user reviews into bugs and feature requests to generate new tickets on task management and online monitoring systems such as JIRA. The proposed methodology involves data preprocessing using data cleansing techniques, classification of user reviews using the Extra Tree set model (trained TFIDF vector), and identification of duplicate reviews using Universal Sentence Encoder, removing duplicate features, classification of bugs and feature requests and generation of issue tickets on JIRA via the JIRA rest API.

Johann, Stanik, Alizadeh B., and Maalej (2017) proposed a simple and consistent approach to retrieve and match the application's features. This approach has three steps. First of all, it identifies POS patterns that determine features. The next step is automatic feature extraction which includes steps such as text preprocessing to remove irrelevant and noisy data and last step is extraction of phrases that are duplicated or have same patterns. The authors evaluated the final results by creating evaluation sets from the features of the app descriptions and reviews, then the results are evaluated by manually extracting the true positives from the features of the two (descriptions and reviews of the applications).

Suprayogi, Budi, and Mahendra (2018) proposed the model for extracting information from reviews using the combination of text mining and sentiment analysis techniques to find implicitly available information in mobile application review. This model consists of several modules: filtering to eliminate "non-informative" reviews, content classification into 4 categories (problem, improvement, question and others) using support vector machine (SVM), Naïve Bayes and logistic regression, Feature extraction using TFIDF, sentiment analysis and Topic analysis using LDA and Non-Negative Matrix Factorization (NMF).

Phong, Nguyen, Pham, and Nguyen (2015) proposed a keyword-based approach for semi-automatic reviews analysis which consists of 3 steps. The first step involves data preprocessing to filter the noisy data, then the nouns and adjectives were extracted using POS tagging, the second step involves keyword recommendation using 3 techniques which include keyword ranking (keywords that often appear with negative reviews), keyword clustering and keyword expansion using the word2vec technique. The third step is to search reviews and analyze trends, which includes extracting user reviews of the products which are most relevant to those keywords. For this task the TFIDF weighting scheme and the vector space model (VSM) were used.

Trupthi, Pabboju, and Narasimha (2016) proposed a technique to improve the extraction and evaluation of features for sentiment analysis. This approach uses a combination of NLP and supervised learning techniques which involves several steps such as identifying the corpus upon which the sentiment features are to be extracted, feature extraction using Bag of Words (BoW), classification by training an ensemble on the training set, then applying to the test set, data preprocessing, eliminating low information features / selecting high information features by calculating the information gain for each word, comparing evaluations.

Pay, Lucci, and Cox (2019) built an ensemble method using three keyword extraction techniques for single documents: Text Rank, RAKE (Rapid Automatic Keyword Extraction) and TAKE. This method combines the candidate keywords from these three techniques and applies the pruning techniques on these keywords and recalculates their scores. Finally, the feature keywords are extracted using dynamic threshold functions.

Htay and Lynn (2013) generated a summary of customer reviews. This paper devised a methodology that follows three steps which includes identifying the features of the product on which the customer commented by performing POS tagging for identification of nouns / noun phrases from reviews, extracting opinion words or phrases by identifying adjective, adverb, verb and noun and identifying orientation and summarizing opinions for each product feature according to their orientation.

Kasri, Birjali, and Beni-Hssane (2019) analyzed the impact of the feature extraction techniques such as BOW, TFIDF and word2vec (Aravec) on the performance of a sentiment analysis using the Arabic language. The extracted characteristics are then evaluated for sentiment analysis using four well-known classifiers such as Logistic Regression (LR), Random Forest, Additional Trees, and SVM. The analysis process in this paper consists of pre-processing the text to turn the text into a consistent form and extracting features by applying BOW, TF-IDF and Aravec, then the results are compared using different classifiers.

Waykole and Thakare (2018) compared techniques for feature extraction such as BOW, TFIDF and word2vec for clinical text analysis. The extracted features are then analyzed using logistic regression and the random forest classifier.

Panichella et al. (2016) proposed and implemented ARDOC: a Java-based classifier for classifying application reviews into problem discovery, feature requests, information giving, information seeking and other by using the combination of natural language analysis, text analysis and classification and sentiment analysis. This approach is almost similar to the approach proposed by Maalej, Kurtanović, Nabil, and Stanik (2016) in which user reviews are classified into four types such as bug reports, user experiences, text ratings and feature requests.

Rose, Engel, Cramer, and Cowley (2010) described RAKE, an unsupervised technique, for extraction of keywords from a single document. This paper compared RAKE with the text Rank technique.

Kowsari et al. (2019) provides an overview of the different techniques for extracting features from a text document, dimensionality reduction techniques, correctly used techniques and algorithms, and evaluation methods and limitations and application of each technique in real world problems. Text and document classification systems are deconstructed in stages such as feature extraction, dimension reduction, classifier selection, and evaluation. This paper indicates that text can be classified into various levels (document level, paragraph level, subsection of sentence level). Feature extraction involves preprocessing data to remove noise, extracting common features by applying TF-IDF, TF (term frequency), Word2vec, GloVe (global vectors for word representation) on preprocessed data.

Onan, Korukoğlu, and Bulut (2016) examined five keyword extraction techniques on classification algorithms based on their predictive performance and ensemble methods for classifying scientific documents (categorization). Authors collected the ACM documents for this study. This paper used techniques such as TFIDF based keyword extraction, more frequent metric-based keyword extraction and keyword extraction based on co-occurrence statistical information, text Rank and eccentricity-based keyword extraction algorithm. Then, an in-depth study of comparison-based learning algorithms (SVM, Random Forest, Logistic Regression and Naive Bayes) is conducted with five widely used ensemble methods (Dagging, Random Subspace, Bagging, Majority Voting, and AdaBoost). The results of different techniques were compared and applied on the collected ACM documents.

Saikia and Singh (2018) analyzed the requirements of the RE documents to classify them as ambiguous or unambiguous. This approach involves data preprocessing to remove noisy data, POS tagging, feature extraction using the BOW technique by considering adverbs, adjectives, modals and determinants.

Shah and Patel (2016) worked on organizing text documents using text classification. The proposed methodology which has four stages: document preprocessing to remove noisy data, selection of features classified as wrappers, filters (used in this paper) and embedded methods, feature extraction via PCA, LSI, clustering methods and classification of the text using decision tree, Naive Bayes, and SVM.

Mehta and Pandya (2020) used different lexicons and ML investigation methodologies and showed their aftereffects. This paper discussed types of analysis, sentiment classification (sentence level, document level and attribute / aspect level), main opinion mining groups (ML approach, lexicon-based method), classification of lexicon-based methods and ML approach. This paper also mentions sentiment analysis resources in the form of blogs / forums, reviews, new articles and social networks. The study shows that ML methods, like Naive Bayes, SVM, and neural networks, have the highest accuracy and can be used as baseline learning methods. Lexicon-based methods can also be very useful in some cases.

Luiz et al. (2018) provide a framework to preprocess, summarize and analyze and explore user reviews for applications in the App Stores. The proposed framework includes data preprocessing, topic modeling using the NMF and SToC method on the output of NMF technique to define the number of meaningful topics in the dataset, sentiment analysis using the SACI strategy which consists of the Sentiment Lexicon and Sentiment Identification and Interface summarization by normalizing the sentiment strength to values from 1 to 5 to match a star rating.

El-Tazi and Lotfy (2020) proposed a multi-layered approach for sentiment analysis. This approach includes a text preprocessing layer, an extraction process layer (POS extraction, Ngram modeling, syntactic chunking), a sentiment analysis layer in which VADER (Valence Aware Dictionary for Sentiment Reasoning), an integration and visualization layer.

Wadera, Mathur, and Vishwakarma (2020) proposed a methodology for sentiment analysis using NLP and various ML classifiers. This paper compared several ML classifiers using a dynamic dataset. The proposed methodology follows data preprocessing, feature extraction using the TFIDF vectorizer, sentiment analysis using text blobs and the use of classifiers for comparison of results such as decision tree, Naive Bayes, etc. Results of classifiers are then compared using recall, accuracy, precision, and F-score and random forest classifier model turned to be most accurate than other models.

Park and Seo (2018) conducted a study on the sentiment analysis on the Twitter corpus. The opinions of user were collected for three AI assistants (SIRI, Google assistant and cortana) from Twitter and VADER were used to classify the sentiments of user opinions into positive, negative and neutral. Each opinion is then quantified to document the matrix and demonstrate statistical significance between groups.

Giatsoglou et al. (2017) proposed a generic methodology that adopts the ML approach to detect the sentiments of the text that expresses the user's opinions in various languages. The approach is proposed to support two methodologies called Model Building and Sentiment Prediction. This technique takes advantage of both word-embedded learning and lexicon-based features (e.g. Word2Vec) to extract hybrid feature vectors.

Singh, Gupta, and Singh (2017) performed sentiment analysis on the Legislative Assembly elections in 2017 and analyzed and retrieved the implications of the tweets collected during the overall duration of the election. This paper proposed methodology that includes data gathering from Twitter, pre-processing, sentiment analysis using TextBlob's Naive Bayes classifier, plotting a chart to show the subjectivity and polarity of data for various political parties.

Kaveh-Yazdy and Zarifzadeh (2020) analyzed news from news channels and Telegram messenger agencies that contain information that users receive during the covid19 pandemic. The process followed includes data preprocessing, topic modeling and monitoring using a two-step framework. In the first step, the Mapper algorithm is used to cluster the sentences and the sentences which belong to the singleton nodes are categorized as miscellaneous sentences. Then TFIDF is used to vectorize the remaining sentences. In the second phase, the topic modeling is performed using LDA. Finally, the list of policies and actions is used to align relevant topics to LDA clusters called topic themes that are implemented by the NCRC.

Banerjee and Basu (2007) analyzed and compared the performance of three topic modeling techniques such as LDA, DCM and vMF mixture models and proposed online variants of vMF, EDCM and LDA. Authors collected a large set of real-world documents in both offline and online settings and showed that LDA is useful for word-level topics and vMF is better for document-level topic clusters. This paper also proposed a practical heuristic for hybrid topic modeling, which offers a good tradeoff between performance and efficacy for text stream processing. Cluster quality was evaluated using nMI and Time, which demonstrated that vMF perform better for batch documents clustering and discovering coherent underlying topics.

Bisgin, Liu, Fang, Xu, and Tong (2020) Applied unsupervised topic modeling techniques to categorize drug with a goal to find topics that group drugs with its similar safety concerns and / or therapeutic uses. First, the labeling sections such as boxed warnings, warnings and precautions, adverse reactions of each drug label were processed by using the Medical Dictionary for Regulatory Activities (MedDRA). Then, topic modeling was performed using LDA to generate 100 subjects and topics were extracted using MALLET. The topic modeling's effectiveness was evaluated by using known information on therapeutic uses and drug safety data. Then group the drugs together using statistical probability.

Calheiros, Moro, and Rita (2017) proposed sentiment classification procedure applied to a specific unit of hotel using LDA to identify opinions on hotel issues and intrinsic relationships between hotel industry and hospitality issues from an eco-hotel choice.

Cambria (2016) discussed importance of Affective computing and sentiment analysis and specified its capability to work as a subcomponent technology for other systems and can enhance their capabilities. As available sentiment analysis methods are capable of mining the explicitly expressed sentences but cannot capture the sentiments that are expressed implicitly so there is a need of automatic tools and technologies that can help in mining the sentiments over the web in real time. This paper explains the common tasks of affective computing and sentiment analysis which includes emotion recognition (extracting labelled emotions), polarity detection (Binary classification of emotions) and multimodal fusion (combining all single modalities into single representation). Multimodal fusion is further divided into two types i.e. feature level and decision level.

This paper also discusses three general categories of affective computing and sentiment analysis that are knowledge based techniques, statistical methods and hybrid approaches and also limitation, weakness and strength of those categories and shows that hybrid approaches work better for affectively computing and analyzing sentiments from given data.

Akhtar, Ekbal, and Cambria (2020) have proposed a stacked ensemble method to predict the degree of intensity for emotions and sentiments by combining the outputs obtained from several deep learning and classical feature based models using a multi-layer perceptron (MLP) network. They have developed three deep learning models based on convolutional neural network (CNN), long short-term memory (LSTM) and gated recurrent unit (GRU) and support vector regression based classical supervised model . The three deep architectures are formed on distributed word representations using Glove and Word2Vec models to learn word embedding. And the SVR model includes a number of features (i.e. Word and Character TF-IDF, lexicon features, VADER sentiment). After combining the outputs, a series of normalizing heuristics are performed to minimize the noise.

Cambria, Li, Xing, Poria, and Kwok (2020) worked on problem of polarity detection from text by integrating top-down and bottom-up learning via an ensemble symbolic and sub symbolic AI tools. They have built a new version of senticNet (common sense knowledge base for sentiment analysis) using deep learning architectures. : Top-down for the fact that it leverages symbolic models (i.e., logic and semantic networks) to encode meaning; bottom-up because it uses sub symbolic methods (i.e., biLSTM and BERT) to implicitly learn syntactic patterns from data.

Basiri, Nemati, Abdar, Cambria, and Acharya (2021) proposed the attention-based CNN-RNN Bidirectional Deep Model (ABCDM) for detection of polarity both long and short user reviews suggested. In ABCDM, GloVe word embedding, bidirectional GRU, bidirectional LSTM, attention mechanism and CNN are used to better capture both long-term dependencies and local features. In this article, ABCDM is also improved using stack generalization, an ensemble method that trains a new model to combine the output of various models already trained on the dataset. The stack generalization assigns different weights to inputs based on different conditions. In the stack generalization algorithm, ABCDM and six basic algorithms are used as level 0 base learners, and logistic regression is used as level 1 meta-learners. This stacked model showed better results than all level 0 models, showing its diversity and power of classification of different sentiment polarity.

Choi, Lee, and Sohn (2017) examined trends in academic research on personal information privacy using Scopus DB. Next, a Document Term Matrix (DTM) is established in which the abstracts will be processed, and then filtering is performed. Then the weighted value of each word is calculated using TFIDF. Subject modeling is then applied using LDA. The optimal number of topics (K) for LDA is determined by maximizing the log likelihood of the topic model. Next, four types of trends (hot, cold, active, and peak) in personal information privacy are identified. The topic name is defined by manual examinations based on previous information and extracted terms.

Du, Kuang, Drake, and Park (2017) propose a fast algorithm that uses divide-and-conquer strategy to calculate NMF called DC-NMF. This paper used an rank-2 NMF calculation algorithm to construct a Binary tree structure of data items and compared DC-NMF with a group of three categories: clustering methods, topic modeling methods and NMF algorithms. Cluster and topic quality is evaluated using normalized mutual information (NMI). For nMI, the calculated cluster membership labels were used as the input symbol set.

Jipa (2018) identified potential categories by exploring the potential value of open-ended responses captured by the survey using NLP techniques with topic modeling. The methodology followed data preprocessing, topic modeling using LDA, data visualization using LDAvis, data exploration using LDA, LSI, HDB alternatives on the same corpus, then the final model was built as LDA subjects- 3 by evaluating using Topic Coherence.

Hassanpour and Langlotz (2015) implemented an unsupervised ML approach to capture the main concepts in repository of radiological reports and partition the reports according to their main foci. This approach modeled radiology reports as a vector in Euclidean space and compared them through a cosine similarity measure. This similarity is used to cluster radiology reports and identify underlying topics from the repository using the k-means clustering algorithm. The evaluation of the technique is performed by running the k-means algorithm for different cluster numbers and the results were verified by domain expert radiologist.

Mifrah and Benlahmar (2020) proposed the technique to compare two unsupervised topic modeling techniques (i.e. LDA and NMF) to determine which technique works best. This paper proposed a parameter for comparing and evaluating unsupervised techniques called coherence. The proposed technique involves pre-processing, extraction of 10 topics for each model (LDA and NMF), and calculation of consistency by c_v measurement of topics for two models, and manual evaluation of the results showing that the LDA model provides more coherent topics than the NMF model.

Mustafa, Zeng, Ghulam, and Arslan (2020) proposed a semi-supervised model for clustering documents in Urdu. The proposed model is a combination of preprocessing techniques, seeded LDA model and Gibbs sampling, called latent seeded Urdu Dirichlet allocation (seeded-ULDA). The existing class labels available in the data are used to evaluate the results of document clustering. The labels are retrieved by using different techniques compared with these existing labels.

Sarne, Schler, Singer, Sela, and Tov (2019) suggest the use of automatic topic modeling corpora of privacy policy using unsupervised learning techniques. This paper proposed approach that includes content acquisition, content cleaning, segmentation, post-processing, and topic modeling using Graph LAB which supports the LDA model. The topics extracted are validated by comparing them with the results reported in previous works using supervised learning (which is highly dependent on manual annotation by experts).

Prendergast (2021) analyzes South Dakota and Nevada slot machine regulations and applies automated NLP techniques to extract and analyze the resulting technical requirements. This paper propose an approach that includes data preprocessing, FR identification by constructing a Naive Bayes model from South Dakota regulations and applied to Nevada regulations

This model predicts functional and non-functional Nevada product requirements from a full set of extracted requirements.

Jha and Mahmoud (2019) present a two-stage study aimed at extracting Non Functional Requirements (NFR) from user reviews available in mobile application stores. This study involves data collection, a two-step process of the NFR extraction process where the first step includes qualitative analysis on the user review dataset sampled from IOS application categories where 250 reviews were manually sampled and human annotators manually examined them to classify them as NFR or miscellaneous and the second phase includes optimizing the dictionary-based multi-label classification approach to capture NFRs in user reviews. The keywords used for the dictionary were the words that the human expert would use to detect NFR in a review.

Kurtanović and Maalej (2017) take up the second challenge on RE17 data to identify the types of requirements using the 'Quality Attributes (NFR)' dataset. This paper investigated how requirements can be accurately and automatically classified as functional and non-functional requirements through supervised ML techniques and discovered how accurately various types of NFR can be identified. The proposed approach involves preprocessing, extraction of feature classification, evaluation of a supervised ML approach using metadata, lexical and syntactic features, and cross-validation of classifiers using precision, recall and F1-metrics based on the SVM classification algorithm.

Lu and Liang (2017) conducted the study of automatic classification of user reviews in Functional Requirement (FR) and NFR. The proposed process includes the collection of user reviews, data preprocessing, manual classification of each sentence into types (NFR, FR or other types), training of classifiers on textual features and their weights obtained by applying BoW, TF-IDF and CHI2, augmentation of user review to take advantage of text semantics. Train classifiers using three ML techniques which are Naive Bayes, J48 and Bagging, and evaluate trained classifiers using precision, recall, and F-measure to analyze classifier performance. Evaluation results showed that augmented user reviews can lead to better classification results, and the Bagging is better suited to grading NFRs than Naive Bayes and J48.

Younas, Wakil, Jawawi, Shah, and Mustafa (2019) proposed an automated approach based on semantic similarity for identifying NFRs from requirement documents. The approach includes a three-step procedure where the first step is to preprocess the data, then the Word2Vector model is trained with Wikipedia and then uses the keyword repository containing the NFR indicator keywords. These pre-classified requirements are generated by the manual classification. The trained model produces better results in terms of NFR identification. The performance of the approach is measured in terms of recall, precision and F-measure. Empirical evidence shows that the semi-supervised automated approach reduces manual human effort in identifying RFNs.

Younas, Jawawi, Ghani, and Shah (2020) proposed a semi-supervised ML method to identify NFR. Authors train the model with the Wikipedia data dump. The proposed approach is described in the multi-step procedure which includes preprocessing, calculating semantic similarity using Word2Vec, measuring the performance of the approach using precision, recall and F-measure.

Material and methods

The purpose of this work is to extract feature requests from online reviews of Mobile Applications of travel industry by considering the four different categories belonging to five different countries. This work focuses on extracting the feature requests/suggestion by using different text mining and opinion mining techniques along with Topic Modeling, Sentiment Analysis, and classification of functional and non-functional requirements. Different techniques used in this study have alse been evaluated using appropriate measures. This section describes the steps performed for this study. Figure 1 shows the overview of proposed model.

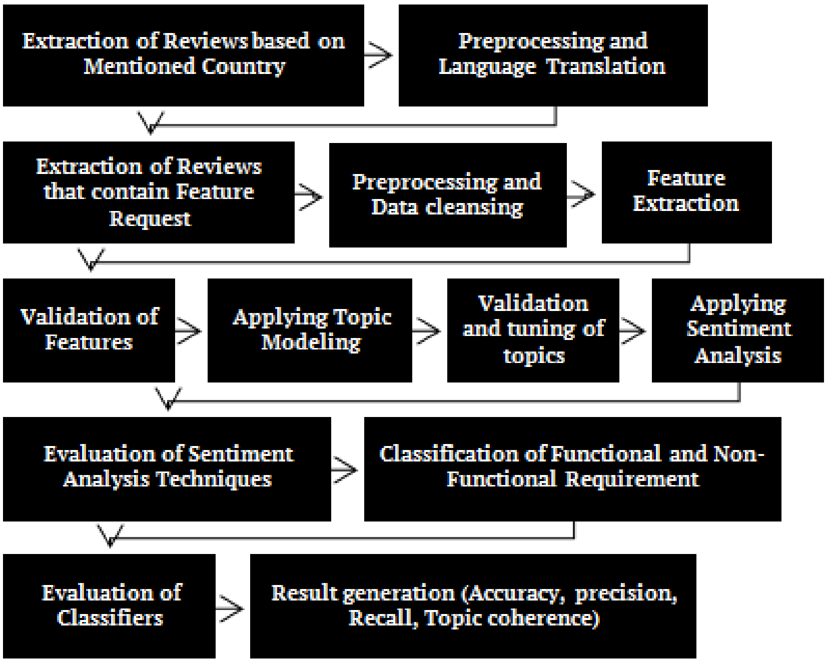
Extraction of Reviews based on Mentioned Country
Preprocessing and Language Translation
Extraction of Reviews that contain Feature Request
Preprocessing and Data cleansing
Feature Extraction
Validation of Features
Applying Topic Modeling
Validation and tuning of topics
Applying Sentiment Analysis
Evaluation of Sentiment Analysis Techniques
Classification of Functional and Non-Functional Requirement
Evaluation of Classifiers
Result generation (Accuracy, precision, Recall, Topic coherence)

Figure 1. Overview of proposed approach.

Extraction of Reviews

To implement the proposed approach, user review data from mobile applications of travel industry considering four different categories belonging to five different countries have been used. However reviews from five applications of each category have been considered. These customer reviews have been extracted from Google play store and Apple store by using App Follow. Table 1 depicts the details of categories and countries of user reviews that have been extracted for this study. The data is collected from January 2010 to March 2020.

Table 1. Apps with categories and countries.

Category	Apps	Countries
Car Booking Applications	1. Careem	1. France
	2. Uber	
	3. Rental cars	2. Russia
	4. sixT	
	5. lyft	3. Korea
Coach Booking Applications	1. busBud	
	2. flixBus	4. Spain
	3. makeMyTrip	
	4. Omio	5. Turkey
	5. Red Bus	
Flight Booking Applications	1. Hopper	
	2. Kayak	
	3. Kiwi.com	
	4. Momondo	
	5. skyscanner	
Train Booking Applications	1. ixigo	
	2. moveIt	
	3. trainline	
	4. trainpal	
	5. wanderU	

Preprocessing and language translation

Since the data collected for the study was in different languages, it was necessary to translate this data into a standard language for further processing. This step consists of three parts.

Removal of emoji's and special characters:

As the data collected was in 5 different languages and each language had different emoji's and special characters, it was necessary to filter out these emoji's and special characters before translating the data because the Google Translation API (used to translate the text into English) does not accepts emoji's or special characters like $, €, MA for translation. These emoji's and special characters were therefore removed from the data.

Language translation:

As mentioned above, the data collected was in five different languages, so it was necessary to translate the data from different languages (i.e. French, Russian, Spanish, Korean and Turkish) in a standard language: English for further processing for feature extraction. The Google Trans API is used to translate text from different languages into English. Google Trans is translation API developed by Google to translate text from one language to another. The Google Trans API supports over a hundred languages and translates text from one language to another with high accuracy.

Pre processing:

As the data has been translated into a standard language, user feedback in this dataset may consist of a word to a paragraph which may contain several sentences. It is possible that in this paragraph, if one sentence contains the user experience and the other sentence contains the user requirement. Since, reviews (sentences) that contain feature requests or user requirements should be extracted. So instead of passing the entire dataset containing paragraphs to retrieve reviews containing feature requests, it was necessary to segment these reviews into sentences. For segmentation of sentences, Tokenize (library provided by NLTK) is used.

Extraction of reviews that contain feature request

As mentioned, user reviews can contain user's feature requests, so it is necessary to extract the sentences that contain those feature requests. Since user reviews can be classified as bugs, user experiences, and feature requests, only feature requests are required for this study. ARDOC (Panichella et al., 2016) has been used to extract sentences that contain feature requests. ARDOC is a Java-based classifier used to classify text sentences. It classifies the text into 5 categories, such as feature request, problem discovery, information seeking, information giving and other. The CSV file of segmented sentences was provided as input to ARDOC and output was a CSV file with its classified categories. Then sentences that have been classified as feature request are separated. Thereafter, data cleaning is performed on extracted feature requests.

Preprocessing and data cleaning

After the segmentation of the sentences, data cleaning was necessary to filter the noise and garbage from the data and get the useful terms used to define the features. NLTK (natural language toolkit) is used for data cleaning. NLTK is a very powerful python package that simplifies all data cleaning tasks. Data cleaning is performed to obtain useful information from reviews and to filter irrelevant data from reviews.

Data cleaning involves the following steps:

Stop words removal:

Stop words are words that are frequently used but are not useful for features requests such as "the", "a", "an", "in" etc. Stop words are removed from the text to filter out unnecessary words from sentences. Since data is in English, NLTK provides a list of stop words of English language that are frequently used. These words contained in the list are removed from the dataset.

Converting text in lower case:

Since the data in the dataset can contain the same words in a different form (upper / lower case), all data is converted to lower case letters to maintain consistency. To convert to lower case, the lower () method of String class is used.

Removal of punctuation:

Since punctuation marks are used frequently by users in reviews and are not useful for feature requests, it is necessary to remove them to include only useful words in reviews. RegexPTokenizer, a library provided by NLTK is used to remove punctuation marks from text.

Lemmatization:

The process of generating the root form (dictionary word) of any word is called Lemmatization. Lemmatization is necessary to as converting the words in the dataset to their root form may increase their count / frequency in the feature extraction technique. If lemmatization is not performed, similar words could be considered as different words and the separate frequency / weight will be calculated for each word in feature extraction techniques. WordNetLemmatizer, a library provided by NLTK is used for this step of data cleaning

Removal of extra symbols:

Symbols like “<”, “*”, “> are also removed from data.

Feature extraction

After data cleaning, the features are extracted. Four methods are used to extract features which are BOW, Rake algorithm, Word2Vec and TFIDF. These methods are widely used to extract features from text reviews. The documents of similar category and country were merged for further processing through these techniques.

In the BOW model, word count of each word is calculated and the word with a high word count is considered to be an important word (i.e. a feature). In this model, the order or words structure in input document is ignored. This model extracts keywords only on the basis of count of words, so it is not a good idea as it can ignore words that are actually features but not frequently used compared to other words in the document.

Therefore, the RAKE algorithm is applied on cleaned data in order to extract features from a given document. RAKE tries to determine the key phrases (using stop words) in a body of text by analyzing the frequency of appearance of the words and its co-occurrence with other words in the text. In RAKE, if a word is frequently used in the document but it is scattered randomly throughout the phrases, it is a less important word and reduces the rank of the sentences in which it appears, on the contrary if the word appears so consistent in the word matches and has fewer total occurrences, so this will increase the ranking of the phrase in which it is found. Therefore, if a word / phrase is used relatively less in the document than other words, it can appear as a keyword using RAKE. Rake can produce possible phrases as keywords and can also produce large phrases if the text does not contain stop words, which is not useful for topic modeling.

So instead of RAKE, the Word2vec model is used to get results that can be used for topic modeling as Word2vec is mainly used for creating words embeddings. Word2vec is a two-layer neural network that processes text by "vectorizing" words. Its input is a text document and its output is a set of vectors: feature vectors which represent the words of this document. Since, in input, the Word2vec model needs any word present in the document and the model returns most similar words of a given word which can be useful to identify the importance of that particular word as feature. As mentioned, Word2vec is good for extracting features, but application stakeholders must first predefine the features to check if those features are the important requirements mentioned in the reviews, which can be a tedious task for application stakeholders.

Thus, instead of Word2vec, the TF-IDF technique was applied to extract features from the text. TF-IDF is the most popular multi document feature extraction method. It computes the importance of the word based on its frequency across all the documents and can provide the keywords which can be features but are used relatively less in corpus. The result of this technique was a single word that can be very useful for topic modeling. For dataset used in this study, TF-IDF works well and gives good results compared to other three techniques.

Topic modeling

After getting the features via TF-IDF, two or more features can be classified under same topic. So to better understand them, Topic Modeling is necessary. Topic modeling is an unsupervised activity that involves making clusters of words with some similarities and each group can be classified as one topic. For topic modeling, two widely used unsupervised topic modeling techniques called LDA and NMF with TF-IDF keywords are used and their results are compared using coherence measure to obtain a technique that works best with user review data. LDA is a topic modeling technique that works on probabilities. It is an unsupervised generative model that provides a stochastic procedure by which the words of documents are generated. Given a corpus of unlabeled text documents, the model discovered hidden topics in the form of distributions on vocabulary words. NMF is a topic modeling technique in which the matrices are constrained to be non-negative. An NMF decomposition of the term-document matrix would produce components that could be considered "topics" and decompose each document into a weighted sum of topics.

Both of these models require a number of topics as a parameters to extract that specific number of clusters. So top 5 clusters of each model have been extracted. To identify the technique which worked best on dataset, c_v coherence measure is used to compare top 10 words of each cluster of each category. High coherence shows that words in cluster are highly relevant to each other. Then results are analyzed and it is identified that LDA clusters are more coherent than NMF clusters. Afterwards, LDA cluster keywords are manually analyzed and each cluster is labelled with most appropriate topic name based on its underlying keywords and clusters with similar topic names are discarded.

Sentiment analysis

Feature Extraction and Topic modeling are good to understand the things that users have discussed in their reviews but to better understand the user’s perspective for any feature, Sentiment Analysis is required. Sentiment Analysis is the process to find user’s opinions. Sentiment analysis categorize the user review as positive, negative or neutral. It shows the stakeholders that which feature got positive responses and are being appreciated by users and which feature got negative responses and are disappointing for users. If the review is neither positive nor negative, it may be categorized as neutral in which user may have just stated his/her experience or knowledge. For sentiment Analysis on user review data, Unsupervised Sentiment Analysis techniques are required. Two well-known unsupervised Sentiment Analysis Techniques named as TextBlob with its two classifiers i.e. Pattern Analyzer and Naïve Bayes Analyzer and VADER are used and their results are compared to find the best technique that work well on this data. For comparing the techniques, Ground truth values are needed that should be used to compare the predicted output of unsupervised sentiment analysis techniques. So the approach used for sentiment analysis follows two steps: 1. Generation of ground truth values, 2. Predicting sentiments by using unsupervised sentiment analysis techniques. In first step, to generate the ground truth values for sentiment analysis, available pre-classified data of reviews which contains sentiments of reviews of mobile applications is downloaded and trained. Four well known supervised ML classifiers (i.e. SVM, Naïve Bayes, and Logistic Regression and Decision Tree classifier) were trained and tested on this pre-classified data. After training and testing these models on pre-classified data, SVM gave 90.36% accuracy for predicting the sentiment of review. This trained SVM model is then applied on user review data to predict ground truth values for sentiments for data used in this study and predictions were analyzed manually as well. In second step, the sentiments of user reviews data used for this study have been predicted by using well-known unsupervised sentiment analysis techniques named as TextBlob and VADER. TextBlob is an open source library available in python that uses the NLTK database to acquire a predefined set of categorized words and works with a rule-based technique. TextBlob has further two algorithms to predict the sentiment of review that are: Pattern Analyzer and Naïve Bayes Analyzer. Pattern Analyzer is a default classifier that is built on the pattern library and Naïve Bayes Analyzer is an NLTK model trained on a movie reviews corpus. These both classifiers of Textblob along with VADER have been applied on user review data to identify the sentiment of feature request. The Textblob shows the polarity and subjectivity of the sentence. The polarity is between [-1,1] where -1 indicates a negative sentiment, 1 indicates a positive sentiment and 0 indicates a neutral sentiment. TextBlob calculates subjectivity using the word intensity. A higher subjectivity shows that the text contains personal opinions rather than factual information. VADER works by following the strategy that uses the dictionary to map lexical features of emotions known as the sentiment score. The sentiment score is calculated by adding up the intensity of each word in the text. VADER produces the four sentiment metrics using these scores which include positive, negative, neutral, and compound values (calculated by normalizing scores for positive, negative, and neutral values).

After analyzing the results of VADER, TextBlob (Pattern Analyzer) and TextBlob (Naïve Bayes Analyzer) are compared with generated ground truth values, it is concluded that TextBlob (Pattern Analyzer) work better than other two for predicting the sentiments by giving relatively better accuracy than others.

Classification of functional and non-functional requirements

User reviews obtained from mobile app stores contain technical feedback that may be helpful to app developers. Since user reviews may contain feature requirements for mobile applications, feature requirements can be classified as functional and Non-functional feature requirements.

Functional requirements are requirements that are explicitly defined by the user and can be implemented as functionality in the application, but non-functional requirements are requirements that cannot be implemented as explicit functionality in the application but the requirements which users expect the application to meet such as better performance, security, ease of use, etc. User reviews can contain any of these requirements. Therefore, classification of the feature request sentences into functional or non-functional requirements must be performed to better understand the user's needs and facilitate the user. Classification of Functional and Non-functional requirements needs to be a supervised activity as each feature request needs to be divided in class so pre-classified dataset of Quality Attributes by re2017 conference is used to train the supervised machine learning models. In this pre-classified dataset, data is categorized into functional (F) and non-functional requirements and the non-functional requirements are then further classified into 11 categories as shown in Table 2. Four supervised ML classifiers (i.e. SVM, Naive Bayes, logistic Regression and Decision tree classifier) have been trained and tested on this pre-classified data and concluded that SVM works better than other classifiers and gives accuracy of 70.74%. This pre-trained SVM classifier model is applied on data used for this study, which classify feature requests into Functional and Non-functional Requirements.

Table 2. Categories of Non-Functional Requirement.

Requirement Type	Class Symbol
Availability	A
Fault Tolerance	FT
Legal	L
Look and Feel	LF
Maintainability	MN
Operational	O
Performance	PE
Portability	PO
Scalability	SC
Security	SE
Usability	US

Experiment results and discussion

As the aim of this work is to facilitate the stakeholders of the application to improve the application based on requirements of users from different countries. So, to collect user reviews from different countries, the Google Play Store and the Apple Store are used.

The data of these reviews was in different languages depending on the country of the user, in order to understand the data and process it further, the data was translated into English language using Google Trans API. Once data for all the applications have been translated in a standard language (i.e. English), the reviews which contain feature requests are extracted using ARDOC. After that, feature request sentences were cleaned using various NLP techniques. After cleaning the data, the features were extracted from the cleaned feature request sentences. To extract feature requests, efforts have been made to compare the results of four different techniques that are used frequently in the literature to find the technique that works best. Those techniques are BOW, RAKE algorithm, Word2Vec and TF-IDF. The results of these techniques were analyzed manually and it was found that TF-IDF has good results and shows the important words as keywords that are more relevant to this study compared to BOW, RAKE and word2Vec. Hence it is concluded from the literature and analysis of results that TF-IDF has good performance for the feature extraction. As TF-IDF can extract feature keywords that are not frequently used but can be important keywords for features

After feature keyword extraction, Topic Modeling is applied to identify the topics as multiple feature keywords can be categorized under one topic. For topic modeling two unsupervised topic modeling techniques i.e. LDA and NMF are used and their performance on mobile applications reviews of various countries and categories is compared using Coherence as shown in Table 3. By analyzing the coherence results, it is concluded that LDA gives better cluster coherence as compared to NMF in most of the cases and LDA clusters were further analyzed manually to label the clusters with topic name according to keywords contained in that cluster. Table 4,5,6,7 shows the topics extracted from mobile applications reviews of different categories for various countries using LDA. By analyzing these topics, it is found that customer requirements vary from one country to other country for similar application as there are different requirements for car booking applications in France than other countries such as Korea, Russia, Spain and Turkey and same goes for other types of applications. So, there is need to include any functionality in application by considering the user requirements of that specific country or region where application is supposed to be used.

Table 3. Comparison of LDA and NMF using coherence.

Applications	Country	Topic Modeling (Coherence Score)	
		LDA	NMF
Car Booking	France	0.5325	0.5108
	Korea	0.5839	0.4438
	Russia	0.4691	0.4790
	Spain	0.5010	0.4013
	Turkey	0.6187	0.6273
Coach Booking	France	0.6035	0.5027
	Korea	0.6047	0.4216
	Russia	0.5844	0.5813
	Spain	0.5672	0.4478
	Turkey	0.5021	0.4065
Flight Booking	France	0.5678	0.4519
	Korea	0.5285	0.4637
	Russia	0.3174	0.3574
	Spain	0.3143	0.3364
	Turkey	0.5965	0.4685
Train Booking	France	0.5239	0.4735
	Korea	0.5241	0.4054
	Russia	0.5301	0.4758
	Spain	0.2494	0.3086
	Turkey	0.4015	0.4481

Table 4. Topics for car booking applications.

Country	Topic Information	
	No.	Topic Name
France	1.	Contact number option in French
	2.	Card number option in French
	3.	Allow pet option for driver
	4.	Call help line for payment problem
	5.	Update payment option chat service
Korea	1.	PayPal account payment option
	2.	Korean language sign In and star rating information
	3.	Driver Login Email and payment option
	4.	Payment and refund option
	5.	Account and Number verification during chat
Russia	1.	Multiple addresses
	2.	Address in Russian
	3.	Preferred address option for driver
	4.	Apple card payment option
	5.	Phone number and preferred color and seat option
Spain	1.	Payment method other than cash
	2.	Cash and card payment in Mexico
	3.	Credit card option in Spanish
	4.	Allow/disallow pet option for driver
	5.	Change card option during payment
Turkey	1.	Improve map of turkey
	2.	Turkish language option in map navigation
	3.	Update location, contact, credit card payment, map and city information option
	4.	Phone number and location option in turkey
	5.	Credit card payment, address and phone number option

Table 5. Topics for coach booking applications.

Country	Topic Information	
	No.	Topic Name
France	1.	Phone network option
	2.	Refund option
	3.	Price, payment currency, Wi-Fi and destination option
	4.	Apple account payment option
	5.	Show bus dissatisfaction
Korea	1.	PayPal payment reservation and live payment comparison
	2.	PayPal payment method in German
	3.	Delete reservation history option
	4.	Accept PayPal payment and refund option
	5.	Card payment option
Russia	1.	City option in Russian
	2.	Destination in map and departure time
	3.	Russian and Ukrainian language and loyalty point option
	4.	Preferred route of city in Russian language
	5.	Chat and seat selection option
Spain	1.	Preferred bus number option
	2.	Preferred customer payment option in Europe
	3.	Add/change language option
	4.	Google map for Bus location
	5.	Student discount and cancel trip option
Turkey	1.	Add lira currency
	2.	Price
	3.	Driver information
	4.	Europe map option
	5.	Contact number

Table 6. Topics for flight booking applications.

Country	Topic Information	
	No.	Topic Name
France	1.	Add, find and download flight information
	2.	Child and price option

	3.	Flight price rate of each day
	4.	Search flight and price information in French
	5.	Favorite flight option in French
Korea	1.	Search and filter price in history by using duration
	2.	Delete history option
	3.	Price and time option during reservation
	4.	Delete ticket and price list
	5.	Date option in iPhone application
Russia	1.	Price option by adding luggage weight
	2.	Add date and price option
	3.	Filter and search flight by using luggage information
	4.	Search flight by route
	5.	Search flight ticket by date and year
Spain	1.	Spanish language option
	2.	Search flight by price
	3.	Add day and price option
	4.	Search flight in Spanish
	5.	Price option in Spanish
Turkey	1.	Add date and month in Turkish
	2.	Send recommendation
	3.	Price in Turkish language
	4.	Time option

Table 7. Topics for train booking applications.

Country	Topic Information	
	No.	Topic Name
France	1.	Strike option
	2.	Flat information option
	3.	Change stop option
	4.	French language option
	5.	Time Option
Korea	1.	Navigate city route feature
	2.	Card information option
	3.	Change bus option
	4.	Show weekend train schedule option
	5.	Travel planning month wise option
Russia	1.	Send arrival information without internet connection
	2.	Track vehicle route option
	3.	City in Russian option
	4.	Show route information on map
	5.	Show available routes in city option
Spain	1.	Route, time and stops information
	2.	Transport route information in application
	3.	Time information according to route in application
	4.	Add preferred time and route
	5.	Transport schedule by using stop information
Turkey	1.	City information
	2.	Search by using time and route
	3.	Send recommendation

Then efforts have been made to use and compare the results of the mostly used unsupervised sentiment analysis techniques to have the technique that gives better results than other two. Those unsupervised sentiment analysis techniques are Textblob (with Pattern Analyzer), TextBlob (with Naïve Bayes Analyzer) and VADER. These techniques are applied on user review data and results are compared as shown in Tables 9, 10 and 11, By analyzing the results, it is concluded that TextBlob with Pattern Analyzer perform well and gives better results as compared to other techniques with accuracy ranging from 75% to 90% on mobile applications of various categories of different countries

After Topic Modeling, sentiment analysis is performed to identify the opinions of users for feature request. As unsupervised sentiment analysis techniques should be used to identify the user's sentiment for feature request. There is a need of truth sentiment values to compare the results of multiple unsupervised sentiment analysis techniques to identify the technique with best results. So to generate Truth values, a pre-classified sentiment data of mobile application reviews is used and a model is trained.

To get the best model to train on this pre-classified data, four commonly used ML models (i.e. Naïve Bayes, SVM, Logistic regression and Decision Tree classifier) are trained and tested and their results are compared as shown in Table 8. By analyzing the results, it is concluded that SVM Model gave high accuracy as 90.38% which is better than other three models (i.e. Naïve Bayes, Logistic Regression and Decision tree classifier) and then the pre-trained SVM model is applied on user reviews (used in this study) to generate truth sentiment values.

Table 8. Comparison of Techniques on pre-classified sentiment dataset.

Classifiers	Accuracy %	Precision %	Recall%
Naïve Bayes	69.23	87.44	41.88
SVM	90.38	87.70	87.23
Logistic Regression	89.38	87.87	84.11
Decision Tree Classifier	89.21	86.60	87.69

Table 9. Evaluation results of sentiment analysis technique - textblob (pattern analyzer).

		TextBlob (Pattern Analyzer)		
Applications		Accuracy%	Precision %	Recall %
Car Booking	France	79.71	82.48	81.77
	Korea	90.32	92.96	91.98
	Russia	83.73	79.48	81.84
	Spain	83.72	81.04	79.39
	Turkey	75.34	71.39	81.79
Coach Booking	France	82.22	83.01	84.39
	Korea	86.67	94.44	77.78
	Russia	88.14	80.85	89.76
	Spain	84.21	79.47	87.38
	Turkey	75.00	73.39	80.00
Flight Booking	France	84.39	82.91	86.12
	Korea	87.35	80.57	85.59
	Russia	82.63	76.14	82.19
	Spain	89.92	83.99	88.47
	Turkey	90.91	82.99	93.25
Train Booking	France	85.26	79.94	83.53
	Korea	86.84	80.95	90.95
	Russia	87.98	83.61	88.19
	Spain	90.31	81.84	87.17
	Turkey	89.63	78.78	86.51

Table 10. Evaluation Results of Sentiment Analysis Technique - Textblob (Naive Bayes Analyzer).

		TextBlob (Naïve Bayes Analyzer)		
Applications		Accuracy %	Precision %	Recall %
Car Booking	France	42.75	14.25	33.33
	Korea	35.48	11.83	33.33
	Russia	28.92	9.64	33.33
	Spain	39.53	13.18	33.33
	Turkey	41.10	13.70	33.33
Coach Booking	France	43.33	14.44	33.33
	Korea	13.33	4.44	33.33
	Russia	33.90	11.30	33.33
	Spain	63.16	21.05	33.33
	Turkey	37.50	12.50	33.33
Flight Booking	France	49.44	16.48	33.33
	Korea	48.80	16.60	33.33
	Russia	45.85	48.57	33.51
	Spain	56.58	18.86	33.33
	Turkey	63.64	21.21	33.33
Train Booking	France	53.21	17.74	33.33
	Korea	52.63	17.54	33.33
	Russia	40.44	13.48	33.33
	Spain	66.78	22.26	33.33
	Turkey	63.95	21.32	33.33

Table 11. Evaluation results of sentiment analysis technique - (VADER).

Applications		VADER Accuracy %	 Precision %	 Recall %
Car Booking	France	44.93	41.94	41.51
	Korea	41.94	47.08	58.29
	Russia	49.40	45.28	49.41
	Spain	54.65	52.12	51.01
	Turkey	54.79	52.36	62.01
Coach Booking	France	66.67	64.00	65.32
	Korea	60.00	65.00	58.89
	Russia	66.10	58.44	61.90
	Spain	56.84	45.95	49.88
	Turkey	59.38	57.14	58.89
Flight Booking	France	62.83	57.00	59.21
	Korea	62.85	58.68	61.21
	Russia	61.61	53.96	57.21
	Spain	69.47	63.23	65.22
	Turkey	76.52	70.04	73.41
Train Booking	France	62.82	55.51	56.07
	Korea	71.05	67.97	70.00
	Russia	65.03	57.50	59.08
	Spain	71.05	57.15	59.91
	Turkey	69.14	58.47	62.67

Comparison with literature

By reviewing the literature, it has been identified that the work done in literature mostly considers user reviews of users for any one or two areas of research or of any one generalized app.

However, the proposed study considers Mobile Application reviews of travel industry applications which includes Car Booking Applications, Coach Booking Applications, Train Booking Applications and Flight booking applications for 5 countries i.e. France, Korea, Russia, Spain and Turkey.

This is the comprehensive study which includes Language Translation, Feature Request Extraction, Topic Modeling, sentiment analysis, Classification of Functional and non-

Functional requirements and Comparison of different techniques used in this study by using appropriate measure of comparison for each task for the above mentioned data. This study also identifies that which technique worked better on given data however, in literature, most of the studies used only one technique for each task and does not give us the proper comparison of available techniques.

This study helps stakeholders to improve their application to meet customer's needs based on their country and they can also have knowledge of user opinions on each feature of their application.

After Sentiment Analysis, Feature Requests were classified into Functional and Non-Functional Requirements. For this classification, four supervised learning models i.e. Naïve bayes, SVM, logistic Regression and Decision Tree classifier were trained and tested on Quality Attributes dataset of re2017 conference and their comparison shows that SVM model gives accuracy as 70.74% that is better than other classifiers as shown in Table 12. This pre-trained SVM model is then applied on mobile application reviews dataset which classify user reviews into 12 categories which are Functional, Availability, Legal, Look and feel, Maintainability, Performance, Portability, Scalability, Security, Usability, Fault tolerant and Operational.

Table 13 shows the comparison of proposed study with other similar Methodologies from literature.

Table 12. Comparison of classifiers on Quality Attributes dataset of re2017 conference.

Classifiers	Accuracy %	Precision %	Recall %
Naïve Bayes	39.89	27.09	12.70
SVM	70.74	70.40	55.52
Logistic Regression	54.26	43.79	28.71
Decision Tree Classifier	51.06	37.60	38.51

Table 13. Comparison of proposed study with literature.

Paper	Data Sample	Methodology	
Suprayogi et al. (2018)	User Reviews	Data Cleansing	NLTK
		Content Classification	SVM Naïve Bayes logistic regression
		Feature Extraction	TFIDF
		Topic Modeling	LDA NMF
Kasri et al. (2019)	ElecMorocco2017 facebook comments dataset	Feature Extraction	BOW TFIDF word2vec (Aravec)
		Sentiment Analysis	Logistic Regression Random Forest Additional Trees SVM
El-Tazi and Lotfy (2020)	500 opinion tweet texts dataset	Feature Extraction	POS extraction Ngram modeling syntactic chunking
		Sentiment Analysis	VADER
Kaveh-Yazdy and Zarifzadeh (2020)	News Information users receive during the covid19 pandemic	Feature Extraction	Mapper algorithm TFIDF
		Topic Modeling	LDA
Akhtar et al. (2020)	datasets of eighth Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis shared task on emotion intensity	emotions and sentiments analysis	Stacked ensemble method by using CNN LSTM GRU support vector regression based classical supervised model
Cambria et al. (2020)	9 different datasets	polarity detection	Top Down (Logic and semantic networks) Bottom up(biLSTM and BERT)
Basiri et al. (2021)	long and short user reviews	Polarity Detetction	GloVe word embedding bidirectional GRU bidirectional LSTM attention mechanism CNN
Du, R. et al. (2020)	text data sets were used which includes Reuters-21578, 20 Newsgroups, Cora, NIPS, RCV1, Wikipedia, Wiki-4.5	Topic Modeling	rank-2 NMF DC-NMF
		Data Exploration	LDA LSI HDB
		Cluster Quality Evaluation	Topic Coherence
Mifrah and Benlahmar (2020)	Corpus of Covid'19 Citations (2019-2020)	Topic Modeling	LDA NMF
		Cluster Quality Evaluation	Topic Coherence
Lu and Liang (2017)	user reviews dataset	Feature Extraction	BoW TF-IDF CHI2
		Classification of FR and NFR	Naive Bayes J48 Bagging
Younas et al. (2020)	Wikipedia data	NFRs Identification	Word2Vector model
Proposed Study	Mobile Application Review data of 5 different	Language Translation	Google Trans API

countries for travel industry	Feature Extraction	BOW RAKE Word2Vec TFIDF

	Sentiment Analysis	VADER
		TextBlob(Naïve Bayes)
		TextBlob(Pattern Analyzer)

	Topic Modeling	LDA NMF
	Cluster Quality Evaluation	C_v Coherence measure
	Classification of FR and NFR	Naïve Bayes SVM Logistic Regression Decision Tree Classifer

Conclusion

This work is done to help requirements engineers, analysts and other application stakeholders to understand customer preferences and requirements in order to improve application in each country where the application is used based on user needs of that country. This helps the requirements engineering process for any application when the intended users of the application are from different countries of the world. This approach shows the feature extraction from user comments available on app stores (i.e. Apple Store and Google Play Store). This study involves the technique to extract features of different categories of mobile applications in the travel industry belonging to different countries. In this approach, a fine-grained analysis is done to extract features from feature requests and cluster similar features under one topic. Then opinion mining is performed on feature request to know the sentiments of users on particular feature request. Afterwards, feature requests are classified into functional or non-functional requirement classes to better understand the user requirements.

This study is carried out by comparing different text mining and information retrieval approaches like BOW, TF-IDF, RAKE and Word2vec to take advantage of approach that work best for extracting features from feature requests. After identifying that TF-IDF can better extract features than other techniques, efforts have been made to assemble similar features into one topic using LDA topic modeling as this gives better cluster coherence than NMF and LDA clusters were labeled manually by analyzing the underlying keywords to identify the topic name for extracted topic clusters. Afterwards, Sentiment Analysis is performed to get the opinion of users for features. As this should be a supervised activity, its truth values are generated by pre-training the SVM model on pre-classified sentiment dataset of mobile application reviews. Then three unsupervised sentiment analysis techniques (i.e. TextBlob with pattern analyzer, TextBlob with Naïve Bayes Analyzer and VADER) are applied on user review data and their results are compared to identify the technique that works best on this data. In the end, SVM, a supervised learning algorithm is trained on pre-classified Quality Attributes dataset of re2017 conference and is applied on mobile applications user review data to classify the feature request into functional or non-functional requirement classes. This is a useful process for application stakeholders as this process can benefit them for getting new features demanded in market or existing features that are required to be improved.

This work presents a comprehensive approach for requirement gathering and feature improvement. In future, this work can be extended by gathering data of other app categories of different countries and from different platforms like Social media sites and other App Stores. Furthermore, this study can also be evolved by using stacked ensemble methods, deep learning and classical feature based models for sentiment analysis and increasing the number of clusters extracted from each topic modeling technique and automating topic name identification process for clusters of similar features.

References

Akhtar, M. S., Ekbal, A., & Cambria, E. (2020). How intense are you? Predicting intensities of emotions and sentiments using stacked ensemble [Application notes]. *IEEE Computational Intelligence Magazine, 15*(1), 64-75. DOI: http://doi.org/10.1109/MCI.2019.2954667

Banerjee, A., & Basu, S. (2007). Topic models over text streams: a study of batch and online unsupervised learning. In *Proceedings of the 2007 SIAM International Conference on Data Mining* [SDM], (p. 431-436). DOI: http://doi.org/10.1137/1.9781611972771.40

Basiri, M. E., Nemati, S., Abdar, M., Cambria, E., & Acharya, U. R. (2021). ABCDM: an attention-based bidirectional CNN-RNN deep model for sentiment analysis. *Future Generation Computer Systems, 115*(3), 279-294. DOI: http://doi.org/10.1016/j.future.2020.08.005

Besrour, S., Rahim, L. B. A., & Dominic, P. D. D. (2016). Investigating requirement engineering techniques in the context of small and medium software enterprises. In *Proceedings of the 2016 3rd International Conference on Computer and Information Sciences* [ICCOINS], (p. 519-523). DOI: http://doi.org/10.1109/ICCOINS.2016.7783269

Bisgin, H., Liu, Z., Fang, H., Xu, X., & Tong, W. (2020). Mining FDA drug labels using an unsupervised learning technique - topic modeling. *BMC Bioinformatics, 12*(11).

DOI: http://doi.org/10.1186/1471-2105-12-S10-S11

Calheiros, A. C., Moro, S, & Rita, P. (2017). Sentiment classification of consumer-generated online reviews using topic modeling. *Journal of Hospitality Marketing & Management, 26*(7), 675–693. DOI: http://doi.org/10.1080/19368623.2017.1310075

Cambria, E. (2016). Affective computing and sentiment analysis. *IEEE Intelligent Systems, 31*(2), 102-107. DOI: http://doi.org/10.1109/MIS.2016.31

Cambria, E., Li, Y., Xing, F. Z., Poria, S., & Kwok, K. (2020). SenticNet 6: ensemble application of symbolic and subsymbolic ai for sentiment analysis. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management* [CIKM], (p. 105-114). DOI: http://doi.org/10.1145/3340531.3412003

Chandankhede, C., Devle, P., Waskar, A., Chopdekar, N., & Patil, S. (2016). ISAR: implicit sentiment analysis of user reviews. In *Proceedings of the 2016 International Conference on Computing, Analytics and Security Trends* [CAST], (p. 357-361). DOI: http://doi.org/10.1109/CAST.2016.7914994

Charrada, E. B. (2016). Which one to read? Factors influencing the usefulness of online reviews for RE. In *Proceedings of the 2016 IEEE 24th International Requirements Engineering Conference Workshops* [REW], (p. 46-52). DOI: http://doi.org/10.1109/REW.2016.022

Choi, H. S., Lee, W. S. & Sohn, S. Y. (2017). Analyzing research trends in personal information privacy using topic modeling. *Computers & Security, 67*, 244–253. DOI: http://doi.org/10.1016/j.cose.2017.03.007

Du, R., Kuang, D., Drake, B., & Park, H. (2017). DC-NMF: nonnegative matrix factorization based on divide-and-conquer for fast clustering and topic modeling. *Journal of Global Optimization, 68*(4), 777-798. DOI: http://doi.org/10.1007/s10898-017-0515-z

El-Tazi, N., & Lotfy, A. (2020). Product features based sentiment analysis from Twitter. *International Journal of Computer Science and Information Security (IJCSIS), 18*(8), 62-73. DOI: http://doi.org/10.5281/zenodo.4012460

Genc-Nayebi, N., & Abran, A. (2017). A systematic literature review: opinion mining studies from mobile app store user reviews. *Journal of Systems and Software, 125*, 207-219. DOI: http://doi.org/10.1016/j.jss.2016.11.027

Ghanyani, U.-S., Murad, M., & Mehmood, W. (2018). Crowd-based requirement engineering. *International Journal of Education and Management Engineering, 2018*(3), 43-53. DOI: http://doi.org/10.5815/ijeme.2018.03.05

Giatsoglou, M., Vozalis, M. G., Diamantaras, K., Vakali, A., Sarigiannidis, G., & Chatzisavvas, K. C. (2017). Sentiment analysis leveraging emotions and word embeddings. *Expert Systems with Applications, 69*, 214-224. DOI: http://doi.org/10.1016/j.eswa.2016.10.043

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

Harman, M., Jia, Y., & Zhang, Y. (2012). App store mining and analysis: MSR for app stores. In *Proceedings of the 2012 9th IEEE Working Conference on Mining Software Repositories* [MSR], (p. 108-111). DOI: http://doi.org/10.1109/MSR.2012.6224306

Hassanpour, S., & Langlotz, C. P. (2015). Unsupervised topic modeling in a large free text radiology report repository. *Journal of Digital Imaging, 29*(1), 59-62. DOI: http://doi.org/10.1007/s10278-015-9823-3

Htay, S. S., & Lynn, K. T. (2013). Extracting product features and opinion words using pattern knowledge in customer reviews. *The Scientific World Journal, 2013*(394758), 1-5. DOI: http://doi.org/10.1155/2013/394758

Iacob, C., & Harrison, R. (2013). Retrieving and analyzing mobile apps feature requests from online reviews. In *Proceedings of the 2013 10th Working Conference on Mining Software Repositories* [MSR], (p. 41-44). DOI: http://doi.org/10.1109/MSR.2013.6624001

Jha, N., & Mahmoud, A. (2019). Mining non-functional requirements from app store reviews. *Empirical Software Engineering, 24*, 3659-3695. DOI: http://doi.org/10.1007/s10664-019-09716-7

Jipa, G. (2018). Mobile applications buying opinions exploration using topic modeling. *Expert Journal of Economics, 6*(2), 44-55.

Johann, T., Stanik, C., Alizadeh B., A. M., & Maalej, W. (2017). SAFE: a simple approach for feature extraction from app descriptions and app reviews. In *Proceedings of the 2017 IEEE 25th International Requirements Engineering Conference* [RE], (p. 21-30). DOI: http://doi.org/10.1109/RE.2017.71

Kasri, M., Birjali, M., & Beni-Hssane, A. (2019). A comparison of features extraction methods for Arabic sentiment analysis. In *Proceedings of the 4th International Conference on Big Data and Internet of Things* [BDIoT´19], (p. 1-6). DOI: http://doi.org/10.1145/3372938.3372998

Kaveh-Yazdy, F., & Zarifzadeh, S. (2020). Track Iran's national COVID-19 response committee's major concerns using two-stage unsupervised topic modeling. *International Journal of Medical Informatics, 145*, 104309. DOI: http://doi.org/10.1016/j.ijmedinf.2020.104309

Kowsari, K., Meimandi, K. J., Heidarysafa, M., Mendu, S., Barnes, L., & Brown, D. (2019). Text classification algorithms: a survey. *Information, 10*(4), 150. DOI: http://doi.org/10.3390/info10040150

Kurtanović, Z., & Maalej, W. (2017). Automatically classifying functional and non-functional requirements using supervised machine learning. In *Proceedings of the 2017 IEEE 25th International Requirements Engineering Conference* [RE], (p. 490-495). DOI: http://doi.org/10.1109/RE.2017.82

Lu, M., & Liang, P. (2017). Automatic classification of non-functional requirements from augmented app user reviews. In *Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering* [EASE'17], (p. 344–353). DOI: http://doi.org/10.1145/3084226.3084241

Luiz, W., Viegas, F., Alencar, R., Mourão, F., Salles, T., Carvalho, D., ... Rocha, L. (2018). A feature-oriented sentiment rating for mobile app reviews. In *Proceedings of the 2018 World Wide Web Conference* [WWW '18], (p.1909–1918). DOI: http://doi.org/10.1145/3178876.3186168

Maalej, W., Kurtanović, Z., Nabil, H., & Stanik, C. (2016). On the automatic classification of app reviews. *Requirements Engineering, 21*, 311-331. DOI: http://doi.org/10.1007/s00766-016-0251-9

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

Malik, H., & Shakshuki, E. (2016). Mining collective opinions for comparison of mobile apps. *Procedia Computer Science, 94*, 168-175. DOI: http://doi.org/10.1016/j.procs.2016.08.026

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. DOI: http://doi.org/10.1109/TSE.2016.2630689

Mehta, P., & Pandya, D. S. (2020). A Review on sentiment analysis methodologies, practices and applications. *International Journal of Scientific & Technology Research, 9*(2), 601-609.

Mifrah, S., & Benlahmar, E. H. (2020). Topic modeling coherence: a comparative study between LDA and NMF models using Covid'19 corpus. *International Journal of Advanced Trends in Computer Science and Engineering, 9*(4), 5756-5761. DOI: http://doi.org/10.30534/ijatcse/2020/231942020

Mustafa, M., Zeng, F., Ghulam, H., & Arslan, H. M. (2020). Urdu documents clustering with unsupervised and semi-supervised probabilistic topic modeling. *Information, 11*(11), 518. DOI: http://doi.org/10.3390/info11110518

Nagappan, N., & Shihab, E. (2016). Future trends in software engineering research for mobile apps. In *Proceedings of the 2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering* [SANER], (p. 21-32). DOI: http://doi.org/10.1109/SANER.2016.88

Onan, A., Korukoğlu, S., & Bulut, H. (2016). Ensemble of keyword extraction methods and classifiers in text classification. *Expert Systems with Applications, 57*, 232-247. DOI: http://doi.org/10.1016/j.eswa.2016.03.045

Panichella, S., Di Sorbo, A., Guzman, E., Visaggio, C. A., Canfora, G., & Gall, H. C. (2016). ARdoc: app reviews development oriented classifier. In *Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering* [FSE 2016], (p. 1023-1027). DOI: http://doi.org/10.1145/2950290.2983938

Park, C. W. & Seo, D. R. (2018). Sentiment analysis of Twitter corpus related to artificial intelligence assistants. In *Proceedings of the 2018 5th International Conference on Industrial Engineering and Applications* [ICIEA], (p. 495-498). DOI: http://doi.org/10.1109/IEA.2018.8387151

Pay, T., Lucci, S., & Cox, J. (2019). An ensemble of automatic keyword extractors: TextRank, RAKE and TAKE. *Computación y Sistemas, 23*(3), 703–710. DOI: http://doi.org/10.13053/CyS-23-3-3234

Phetrungnapha, K., & Senivongse, T. (2019). Classification of mobile application user reviews for generating tickets on issue tracking system. In *Proceedings of the 2019 12th International Conference on Information & Communication Technology and System* [ICTS], (p. 229-234). DOI:http://doi.org/10.1109/ICTS.2019.8850962

Phong, M. V., Nguyen, T. T., Pham, H. V., & Nguyen, T. T. (2015). Mining user opinions in mobile app reviews: a keyword-based approach (T). In *Proceedings of the 2015 30th IEEE/ACM International Conference on Automated Software Engineering* [ASE], (p. 749-759). DOI: http://doi.org/10.1109/ASE.2015.85

Prendergast, M. D. (2021). Automated extraction and classification of slot machine requirements from gaming regulations. In *Proceedings of the 2021 IEEE International Systems Conference* [SysCon], (p. 1-6). DOI: hhtp://doi.org/10.1109/SysCon48628.2021.9447144

Rose, S., Engel, D., Cramer, N., & Cowley, W. (2010). Automatic keyword extraction from individual documents. In M. W. Berry & J. Kogan (Eds.), *Text Mining: Applications and Theory* (p. 1-20). Hoboken, NJ: Wiley.

Saikia, L. P. & Singh, S. (2018). Feature extraction and performance measure of requirement engineering (RE) document using text classification technique. In *Proceedings of the 2018 4th International Conference on Recent Advances in Information Technology* [RAIT], (p. 1-6). DOI: http://doi.org/10.1109/RAIT.2018.8389074

Sarne, D., Schler, J., Singer, A., Sela, A., & Tov, I. B. S. (2019). Unsupervised topic extraction from privacy policies. In *Proceedings of the 2019 World Wide Web Conference Companion* [WWW´19], (p. 563-568). DOI: http://doi.org/10.1145/3308560.3317585

Shah, F. P., & Patel, V. (2016). A review on feature selection and feature extraction for text classification. In *Proceedings of the 2016 International Conference on Wireless Communications, Signal Processing and Networking* [WiSPNET], (p. 2264-2268). DOI: http://doi.org/10.1109/WiSPNET.2016.7566545

Singh, A. K., Gupta, D. K., & Singh, R. M. (2017). Sentiment analysis of twitter user data on Punjab legislative assembly election, 2017. *International Journal of Modern Education and Computer Science, 9*, 60-68. DOI: http://doi.org/10.5815/ijmecs.2017.09.07

Sohail, S. S., Siddiqui, J., & Ali, R. (2016). Feature extraction and analysis of online reviews for the recommendation of books using opinion mining technique. *Perspectives in Science, 8*, 754-756. DOI: http://doi.org/10.1016/j.pisc.2016.06.079

Suprayogi, E., Budi, I., & Mahendra, R. (2018). Information extraction for mobile application user review. In *Proceedings of the 2018 International Conference on Advanced Computer Science and Information Systems* [ICACSIS], (p. 343-348). DOI: http://doi.org/10.1109/ICACSIS.2018.8618164

Trupthi, M., Pabboju, S., & Narasimha, G. (2016). Improved feature extraction and classification — Sentiment analysis. In *Proceedings of the 2016 International Conference on Advances in Human Machine Interaction* [HMI], (p. 1-6). DOI: http://doi.org/10.1109/HMI.2016.7449189

Wadera, M., Mathur, M., &Vishwakarma, D. K. (2020). Sentiment analysis of tweets- a comparison of classifiers on live stream of Twitter. In *Proceedings of the 2020 4th International Conference on Intelligent Computing and Control Systems* [ICICCS], (p. 968-972). DOI: http://doi.org/10.1109/ICICCS48265.2020.9121166

Waykole, R. N., & Thakare, A. D. (2018). A review of feature extraction methods for text classification. *International Journal of Advance Engineering and Research Development (IJAERD), 5*(4), 351-354.

Younas, M., Jawawi, D., Ghani, I., & Shah, M. A. (2020). Extraction of non-functional requirement using semantic similarity distance. *Neural Computing and Applications, 32*, 7383-7397. DOI: http://doi.org/10.1007/s00521-019-04226-5

Younas, M., Wakil, K., Jawawi, D. N. A., Shah, M. A., & Mustafa, A. (2019). An automated approach for identification of non-functional requirements using word2vec model. *International Journal of Advanced Computer Science and Applications (IJACSA), 10*(8). DOI: http://doi.org/10.14569/IJACSA.2019.0100871