Journal of Information Systems Engineering and Management
2025, 10(16s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Classification of Non-Functional Requirements Using Semi-Supervised Learning Approach

Devendra Kumar[1] , Anil Kumar[2] , Laxman Singh[3] ,

[1]Research Scholar, Dr A.P.J. Abdul kalam Technical University, Lucknow, U.P., India.

Email: kdevbali@gmail.com

[2]Department of Computer Science, B.I.E.T. Jhansi, U.P., India, Email: solankibiet13@gmail.com

[3]Department of Electronics &amp; Communication, Noida Institute of Engineering and Technology,Greater Noida, U.P., India.

Email: Laxman.mehlawat2@gmail.com

Corresponding Author: kdevbali@gmail.com

ARTICLE INFO

Received: 14 Dec 2024

Revised: 28 Jan 2025

Accepted: 10 Feb 2025

ABSTRACT

The primary emphasis is on specialized work involving software engineers. Nevertheless, it is equally imperative to document the deficient features of software design, including aspects such as maintainability, reusability, and reliability. In rapid software development methodologies, such as agile, non-functional requirements (NFRs) are frequently neglected. This neglect results in an increased significance of eliminating non-functional requirements in agile-based software and a heightened emphasis on critical tasks during software migration. Misinterpretation of NFRs can be a significant factor contributing to project failure. A computer network simplifies the implementation of 12 key concepts such as NFR heuristics in the context of failing rules. We propose a semi-written classification system to identify useless needs. In this manner, the initial distribution forthe NFR is learned using the small set determined during the heuristic phase. This iterative process helps identify additional needs. The aim was to incorporate this approach into individual recommendations to assist analysts and software designers in the architectural process. Using a semi-supervised learning method, NFRs can be effectively identified and classified. In addition, usingother information provided by well-written and informal rules allows classification using fewer preexisting methods. The learning process improves the classification performance by leveraging user feedback during training. Our partial maintenance strategy provides over 70% accuracy compared withtraditional maintenance using pre-recorded data. This demonstrates the superiority of the semi supervised approach. The researchers also noted that partial care systems require less human effortto enroll than full care and could be further developed according to the guidelines. Currently, ourmethod is better than other distribution tracking methods, and we believe it will improve the performance of collaboration with input from analyst participants.

Keywords: Artificial Intelligence, Financial Marketing, Customer Segmentation, Risk Assessment, Machine Learning, Big Data Analytics, Predictive Modeling, Credit Scoring, Fraud Detection, Robo-Advisors, Personalized Marketing, Algorithmic Bias, Data Privacy, Fintech Innovation, Regulatory Compliance.

INTRODUCTION

Non-functional rules (NFRs) play an important role in the design and development of software systems. These requirements define the desired properties of the final system. NFRs include security, speed, availability, scalability, and portability, which are all related to software products. NFRs should be defined early in the architectural design process for inclusion in the design considerations. However, this process can be difficult and time-consuming because most NFRs are often included in business requirements and are therefore easily overlooked. In recent years, methods such as surveys, checklists, and models have become popular to remove dysfunctional rules from participants [1, 2]. In addition, the resulting process can be further classified and developed using extended vocabulary or specialized knowledge such as ontologies[3]. These processes have proven effective in increasing customer satisfaction, adapting to changing needs, facilitating modern software development, and directly improving communication with customers. Cloud computing has also emerged as a cost-effective software development solution that provides storage and processing power and reduces the installation and upgrade time. Various cloud services, including data sharing, infrastructure, distributed application development, and core functionality, help to reduce software development costs [4] and [5]. Unfortunately, agile software development methods often ignore non-functional requirements (NFRs) as they are more important than functional requirements (FRs) during design and implementation. NFR can be poorly communicated or misunderstood during agile development, often owing to a lack of user experience. While NFR terminology and categories are based on good standards [6], integrating NFR heuristics into agile processes poses challenges. Various NFR heuristics, such as models [7,8,9] exist, but most are designed for traditional software development environments. Cloud computing platforms such as Team Foundation Server, Microsoft SharePoint, and Pivotal Tracker provide options for the engineering needs of agile-distributed software development. However, employees must have a good understanding of writing NFRs, and unfamiliarity with NFRs can exacerbate this problem. Therefore, new methods are required for obtaining NFR data. The lack of NFR heuristics in agile processes has been criticized [10]. Previous studies have demonstrated the benefits of NFR aggregation strategies that use historical data to predict the future NFR and FR in agile software development. The cloud environment facilitates the debugging process of NFR by enabling the exchange of information between team members in different locations. For example, tools such as Google Docs have been used to access NFR documents stored by previous team members [11]. In this study, we use the sales method to define NFRs [12].

Our Methodology aims to describe and classify NFRs in an agile model. The aim was to reduce the number of required learning styles. The main contributions of this study are as follows.

1. We used natural word associations in the text by combining labeled and unlabeled data, which resulted in significant improvements in learning accuracy.
2. The results from the distribution can provide an initial set of needs, obtained through interviews or other methods, that analysts can use to identify needs. Analysts can obtain recommendations to categorize the remaining needs and provide feedback to improve the categorization.
3. Section 2 discusses previous related work in this area, and Section 3 discusses semi-supervised learning, which is the motivation behind the ML techniques used in this study.
4. Section 4 (from 4,1 to 4.3) discuss the related work in which every step from preprocessing to the classification and used algorithm ie EM classifier is described
5. Section 5 discusses the results and analysis of the proposed methods using comparative tables and graphs.
6. Finally, Section 6 concludes the paper and describes future work.

BACKGROUND OF THE STUDY

Semi- supervised Learning

Figure 1: Semi-Supervised Learning Approach [23] Proposed Approach eight using cloud communication technology. These platforms allow for the sharing of codes, information, and ideas, facilitating rapid feedback and collaboration. Team collaboration is supported by a variety of methods and tools [24, 25]. Cloud storage systems play an important role in sharing information needs by providing rapid access to information, facilitating the exchange of ideas, providing forums and wikis, and enabling real-time collaboration through shared capabilities. It also provides a software as a service (SaaS) approach, project management, code hosting, and software testing tools. By leveraging the cloud, agile teams can access specific data from multiple sources to improve the extraction of non-functional requirements (NFRs). After the request has been evaluated, it can be sent to the group root server of the broadcast center. The plan uses the EU e-call dataset as the dataset

This study aims to explore the role of AI in transforming customer segmentation and risk assessment in financial marketing. By analyzing current trends, opportunities, and challenges, this research will provide insights into how AI-driven strategies can enhance financial decision-making while addressing ethical and regulatory considerations.

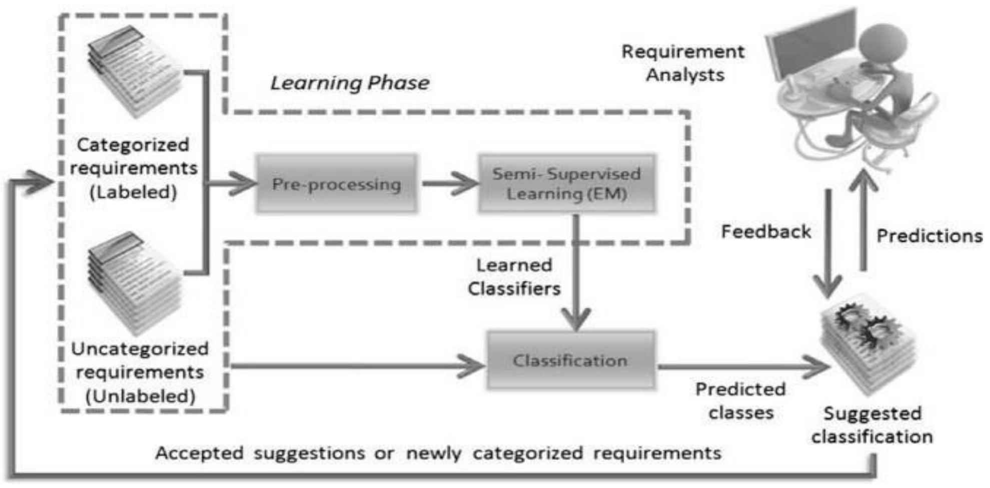
Learning Phase
Requirement Analysts
Categorized requirements (Labeled)
Pre-processing
Semi- Supervised Learning (EM)
Feedback
Predictions
Learned Classifiers
Uncategorized requirements (Unlabeled)
Classification
Predicted classes
Suggested classification
Accepted suggestions or newly categorized requirements

Figure 1: Semi supervised learning approach [23]

LITERATURE REVIEW

In the early stages of software development, it is common practice to explain simple English with informal explanations. In recent years, automated methods have been developed to help manufacturers check necessary information; however, the results are different [13]. Information retrieval (IR) and natural language processing (NLP) techniques have been used to create systems that facilitate (semi) semi-automated analysis. Park et al. [14] proposed a support method that uses similarity measures from recovered data to identify similarities, differences, and uncertainty. Another system called ReqSimile [15] assists this process in seeing synthetic materials by creating links that can follow the query retrieval method, connecting the requester to the required product. ReqSimile evaluates the probability that a new requirement is related to an existing requirement by comparing its functionality and content with the existing requirements. TTC (occasional integration) is used to categorize and index specifications as they are helpful when needed. The Reuse Assisted Requirements Elimination (RARE) method uses domain mapping synonyms and NLP techniques to analyze and refine requirements to create a semantic network [16]. English skills were also used to assess the text quality. A simple structuring process can identify simple errors and complex issues that human observers may miss, whereas a process-based approach helpsidentify inconsistencies in the expectations of many participants [17]. Fantech et al. [18] exploredthe use of speaking tools to collect quality measures and explore problems caused by ambiguity inexplanations. Natural language processing has been used to bridge the gap between formal and informal rules, thus providing a solid foundation for in-depth analysis. A web-based analytical support system was proposed for classification [20]. To obtain an informal classification of needs according to different perspectives or problems, the system needs input from each analyst's perspective. Based on the integration of the points in

the demand list, the system begins to expand, and point centers are created to classify new demand. However, this approach does not allow analysts to rely on obtaining information about the individual components of the program under development. In the search for uncertainty, classification algorithms are used to create a system that uses quality measures from a previous model [21]. Recently, non-functional regulations (NFRs) have been reviewed by monitoring departments. NFR classifiers are trained using an iterative process to generate weights for each NFR category (safety and performance). The frequency of relevant statements in the text should be used to estimate whether a new requirement will meet a particular NFR. Even taking into account the differences in the selection process, NFR classifiers outperform existing classifiers such as Naive Bayes and Traditional decision trees. An automatic speech recognition system was developed to record NFRs during meetings with participants using unstructured text [22]. However, the detection of NFR using control methods requires many preliminary classification procedures, making classification difficult. NFR classifiers can identify and classify specific needs in ongoing projects and use data from previous projects; however, this approach can be difficult because of the many languages, item spellings, and spellings across projects. Semi-Supervised Learning Complying with non- functional requirements (NFRs) presents significant challenges for software analysts and designers due to the unique nature of these requirements. NFRs often describe the characteristics of a customer's needs, and it is important to find, identify, and monitor key features in the required information. Analysts working on software development projects need to review the necessary data to determine if the requirements are working, make decisions about the product, criteria, or categories of NFRs (e.g., security, usability, and scalability), and be able to prioritize. and architectural decisions, respectively. The process of identifying and analyzing each document takes time and requires human reading and categorization. An ad hoc text-tracking system based on machine learning and artificial intelligence was developed to solve the problems associated with the identification of non-functional requirements. This approach uses discrete electronic components to distinguish between different types of NFRs and provides these differences to analysts. The system attempts to gather information, each of which explains the requirements in English [23]. Figure 1 provides a high-level overview of the planning process. Initially, automated tags were trained using semi-supervised learning techniques that use both nonprescriptive rules and classification models. To achieve this goal, expectation-maximum (EM) techniques combined with Naive Bayes classifiers were used. The team can discover the initial classification of user needs during user conversations. Additionally, the set of predicates specified in [8] can be used as a taxonomy system, providing a simple approximation. Once the classifier is trained and developed, it can be used to classify systems that have not been previously described. In some cases, fidelity needs or feedback from analysts can be reused as the needs of the iterative process, thus completing the process [24].

MATERIAL AND METHODOLOGY

Proposed Method

The proposed agile software development process is based on customer engagement, teamwork, and multi-stakeholder collaboration and is facilitated by the cloud computing communication platform. These platforms allow for the sharing of codes, information, and ideas, facilitating rapid feedback and collaboration. Team collaboration is supported by various methods and tools [25]. Cloud storage systems play an important role in sharing information needs by providing rapid access to information, facilitating the exchange of ideas, providing forums and wikis, and enabling real-time collaboration through shared capabilities. It also provides a software as a service (SaaS) approach, project management, code hosting, and software testing tools. By leveraging the cloud, agile teams can access specific data from multiple sources to improve the extraction of non- functional requirements (NFRs). After the request has been evaluated, it can be sent to the group root server of the broadcast center. The plan uses the EU e-call dataset for this study [26].

Preprocessing

The vector space model (VSM) is widely employed in information retrieval (IR) and machine learning to construct effective text representations. In this model, each document is represented by a feature vector in a multidimensional space, where each dimension corresponds toa specific word and the numerical value or weight reflects the importance of that word in the document [27]. By assigning a unique vector wij to each document in collection D, the weight of a word in document DJ can be determined by only examining that document. This is shown in (1).

DJ’s vector we.:

$$d = (w_{y_1} w_2, \dots, w_1) \tag{1}$$

“In the following equation, each term t 2 T, where jTj indicates the total number of distinguishing words in the document collection or vocabulary, is weighted using the term fre- quency–inverse document frequency (TF–IDF) [39] term weighting function, which is presented as follows:”

$$= i([t_i, d_j) = \hat{N}(t_i, d_j) \times log_j \left(\frac{-9|}{\|_0(t_i)} \right) \tag{2}$$

The equation uses the letter Ti to represent both a term and document, whereas dj represents a document. #ti indicates the frequency of a term in the collection, and D represents thetotal number of documents. The total number of documents in D is denoted by jDi and the frequency of documents appearing in D is denoted by #dj. Ti represents the frequency of the term occurring in D and jDi represents the total number of documents in D. Alternatively, ti can be used. This metric combines two field observations, with the more frequent occurrences of a phrase being increasingly important for recognizing the type of document in which it appears. There aretwo reasons for this: first, as phrases are used more frequently, they become less efficient at distinguishing between different articles; and second, as a phrase is used more frequently, it becomes less effective at distinguishing various articles. To transform textual requirements into vectors based on the vector space model, several preprocessing steps must be performed.

When processing redundant data, several preliminary procedures are typically used, including optimization (replacing letters, numbers, and hyphens) and root extraction (removal of stems), which is a combination of the previous two [28]. The first step in the process is to shrink the desired data by removing numbers or sentences containing numbers, separating hyphens to form new words, and converting all letters to numbers (as described earlier). After removing the stop message, the previous step continues. Stop words, such as the preposition "moment", are words ina language that often help form sentences or phrases, but have little or no meaning. These words have many forms such as objects, prepositions, conjunctions and pronouns. Most abandoned words are removed using predefined lists or negative dictionaries containing terms that do not distinguishbecause of their frequency or meaning. Using the English stop list [29], the test included 319 different items. After removing the stops, stem subtraction is used to reduce the morphological variation from the stems of the remaining words. This includes dealing with multiple nouns, verbtenses, and gerunds. Stemming is a language normalization that reduces the morphological conjugation of words to their roots (stems) using mathematical equations. Combining morphologically related words into a recursive root helps reduce the size of the vector space, and Porter's method [30] is used to extract the content of the text with the method. The nextpreprocessing step involves applying the tf-idf weights defined in Equation 1 to all generated content (real bodies) [2]

Bayesian Classification

Supervised learning is a method of creating a classification system that can predict labels for new events based on previously taught models (also called reinforcement learning). In the literature, the terms "education", "classification" and "classification" are frequently used todescribe the educational process. It is defined as one or more classes or groups, classes or groups, all of which must contain data with certain characteristics, and all must have certain characteristics.

Create an initial instruction or model to track learning throughout the training phase. A previouslycreated classification or pattern is used for new data to determine which category they belong to during classification. The classification step uses a supervised learning method to predict the class of new data. Studies have shown that the Naive Bayesian text classification algorithm can be used to classify text effectively [29]. To classify new events, we can use Bayesian methods to help us identify the most common classes based on available data. According to the Bayesian model, the dataset should be generated from a parametric model where training data is used to provide good Bayesian estimates of the model parameters. New test data are classified using Bayesian rules according to their probability of forming certain data due to these calculations [31]. The categorization procedure is considerably easier to understand and carry out if you select the mostlikely class.

For text classification applications, the multinomial model has been found to be the most commonly employed naive Bayes classifier. Documents are constructed using a mixed model in which the model's vocabulary is used to generate an ordered list of terms for each document. Hereare some examples of how to make use of the fourteenth power: hw1. Based on the training data, the symbol H is used to indicate the expected value of H, which is denoted by Hb. When comparingthe frequency of a given word in training data Dj to the frequency of the same word in all training data for a specific class (e.g., weight/class), the word frequency ratio is determined. Using Laplacian smoothing, you should be able to handle zero counts for uncommon terms as follows:

$$(w_t \mid, \hat{\theta}_j) = \frac{1+\sum_{d=1}^{N_\sharp} (c_j \mid d_j)}{|V|+\sum_{n=1}^{N/2} \sum_{i=1}^{Nu} (c_j|d_i)} \quad (3)$$

The class prior probabilities can be also calculated using training data as follows

$$(c_1 \mid \hat{g}) = \frac{1+\sum_1 (c_i d_i)}{|C+|s|} \quad (4)$$

The class prior probabilities can be also calculated using training data as follows

More informally, Eqs. (3) And (4) can be re-written as:

$$P(w_i \mid, \hat{B}_j) - \frac{1+ \text{No. of occurrences of } w_2 \text{ in class} \downarrow}{|Y+ \text{No of wonds inclass } i} \quad (5)$$

+

$$P(c_j \mid \theta) = \frac{1+ \text{No. of documents in dass } f}{|C|+|?|} \tag{6}$$

Estimates of both parameters produced from training documents must be utilized in conjunction with each other in order to classify test documents. Predicting the most likely class based on the data in all tests and then choosing the test with the highest posterior probability fulfills this objective. This can be represented as follows using the Bayes rule:

$$(c_1 \mid \theta) = \frac{1+ \text{No. of documents in dass } |}{|C|+|9|} \tag{7}$$

$$= \frac{(c_j|\hat{\Theta}) \prod_{k=1}^{d_1} P(w_{d,k}|c_j}{\Sigma^{C}[\ (c_r|\bar{\Theta} \quad _{k-1} (\text{ though } w_{di,k}|c_r;\bar{\Theta} \quad r = \prod^{|d|} \quad 1} \tag{8}$$

Naive Bayesian classification is based on the assumption of class conditional independence, which is established to decrease computation when evaluating Pdijcj; Hb: in order to reduce computation while assessing Pdijcj. The above equation's substitutions are explained by the assumption that words in a class are conditionally independent of one another. Applying Bayes theorem to each candidate class in C and calculating the posterior probability of each candidate class in C yields the MAP hypothesis, also known as the most likely class hypothesis. cMAP is chosen in one of the following methods for each possible class cj:

$$c_{hm} = arg\ \ max: (c/\underline{u}_i; \theta^{-}) \tag{9}$$

Classification

If you choose the best class, the classification process is easier to understand and do. For text classification, the polynomial model was found to be the most commonly used Naive Bayesian classifier. The document is created using a composite model using standard language to generate a script for each document. Here are some examples of using the fourteenth power: hw1.According to the training data, the expected value of H is represented by the symbol H, which isHb. The term frequency ratio is determined by comparing the frequency of a particular word in the Dj dataset with the frequency of the same word in all datasets for a class (eg weights/groups).Using Laplacian smoothing, you should be able to calculate the number of rarefactions as follows:registration and selection methods are used to generate "distributions or hypotheses. (e.g. Naive Bayes, k-NN, etc.) can rely on signs for new events, this method requires a lot of registration. It isalso expensive and time consuming to write, and is also a mistake as it has to be done manually by business professionals [32]. A good example of partial maintenance is the classification of documents. Semi-supervised learning is a method of partial supervision that involves learning from registered and unregistered documents or information. Sometimes this course is called the LU course (L for registered, U for unregistered). The Iterative Algorithm for Maximum Value Evaluation (MLAE) includes the Expectation-Maximization (EM) method with a threshold and a maximum level. The expectation maximization (EM) algorithm consists of two steps: the primingstep (also called the E step) and the maximization step (sometimes called the M step). Its core consists of basic and advanced levels. This means that in the first stage it uses the current estimateto help fill the data gap, and in the second stage it re-estimates the index to increase the probability.Unnamed files can be considered incomplete because they do not contain class tags. The process is repeated as necessary to obtain local minima and unstable structures. After step M, script is used to start step E, which continues forever. In this way, the main algorithm can be repeated once until the desired result is achieved. The EM algorithm for EM learning using the Naive Bayes distribution described by Nigam and colleagues is presented in Principle 1 as the EM algorithm for LU learning using the Naive Bayes distribution. Method 2 here presents an EM approach to learning EM using Naive Bayes distributions. Therefore, EM should be used to estimate the probability that each word formed in the Bayesian equation (3) and (4) belongs to the class, and the previous probability for each class, respectively. Thus, the L index contains the class index, but the unsigned U index does not have the class index. This much. List L contains the list of names of all files in it. In order to use the current model to assign a label to the class that will existfor each document with a U classifier, missing labels need to be estimated from the current model performed by the EM algorithm. Iteration after iteration, each form in U is given a probability distribution of the class it can go to, such as Pcjjdi, the probability distribution takes values â€‹â€‹in the range of 1/(20/1). A class ck is already known, while the data in L are members of various previously unknown classes. Naive Bayes classification is used to combine two datasets by combining registered and unregistered data and the Pcjjdi distribution into a single distribution.The process is then repeated until Pwtjcj and Pcos become unstable or change slightly, at which point it ends. The main theory of EM states that there is a one-to-one relationship between the components and the members of the class represented by a component in the compound. This is equivalent to claiming that a group can be divided into several subtopics, and if this assumption is violated in the literature, each topic is better defined by dividing it into different words. This assumption is valid enough because NFRs focus on specific software rather than covering a wide range of issues. We try to evaluate the system by using ourselves according to the work and non- work data coming from the field.

Algorithm 1. EM algorithm with naïve Bayesian classificationInput:

Labeled set L of Data D_j, Unlabeled set U of D_i

Output: Classifier f

```
1: Learn an initial naive Bayesian classifier f using only the labeled set L (using Equations (3)and (4)).
2: Repeat
3: For each example di in the unlabeled set U:
4: Compute P(c|di) using the current classifier f (using Equation (7)).5:
End for
6: Learn a new naive Bayesian classifier f from the combination of labeled set L and unlabeledset U by
computing Pr(c) and Pr(w|c) (using Equations (3) and (4)).
7: Until the classifier parameters reach stability.8:
Return the classifier f from the last iteration.
```

RESULTS AND DISCUSSION

Our aim is to evaluate the performance of the semi-supervised requirements classification, our study compares with the findings from the previous set of control criteria to evaluate the methods proposed in this study based on experimental design, results and comparisonswith others methods. In this study, a semi-supervised text classification strategy (NFR) will be explained to analyze ambiguous sentences. Therefore, the effectiveness of such products (especially those using the expectation-maximization method) when distributing paper should be important for most to discover. Also, we take Rocchio's method, k-NN andNaive Bayes and TF-IDF classifier for comparison, and we usually use k-NN and Naive Bayes weights. There are many software qualities required for a system to function properly, such as accuracy, speed, security, and flexibility. NFR focuses on these and other aspects of software quality, including security. To account for the many negative features to consider, N binary classifiers must be trained, each of which is trained to distinguish what is required in one class from what is available in all classes. To train the classifier, the data contained in the class is used as positive and negative examples for other binary classifiers. Each document is tested for classification using N classifiers and the classifier with the highest value or efficiency gets the label with the query class. Multiclass binary classification uses the "many versus many" approach, called the multiclass binary classification method. The has more full successes and less partial successes and failures using the NFRElicit process than the previous version. Three factors were used to assess effectiveness and are further explored in the research literature: comparison of NFR vs. success, partial success and failure of proposals; Comparing the success, partial success and failure of NFR accreditation; and compare artifacts and features used in scenarios to arrive at final conclusions. Table 1 compares the total NFR success shown in the required column. In table 2the results of the x-axis and the number of successful NFRs are shown in the instructions to be given on the y-axis. However, only 89.57% of root NFRs can be determined using the heuristic, whereas 80.70% of root NFRs can be determined using theNERV method and 87.31% using the CEP method. This is an improvement of 8.77% and1.76% over the NERV and CEP methods, respectively. For example, the figure below shows a comparison between a partially completed NFR and a fully completed NFR in therequested sentence. When using a method, it shows on the x-axis the number of messagesrequired for partially completed NFRs and the percentage of messages required with partially completed NFRs; If path is not used, some achievements will not show in the y- axis NFR. The method detecting 8.77% of baseline NFRs compared to 15.79% for the CEP method and 15.79% for the NERV method, significant improvements have been made in reducing the number of successfully NFR-identified sentences in sentences compared to the previous process, NERV method correctly identified 7.02% and 1.75% of the baselineNFR.

Table 1: Baseline Non Functional Requirements

Baseline Non-Functional Requirement		
NFRs	**Labeled Words**	**Status**
Accessibility:	“administrator =	Success
	choose =	Success
	lhcp =	Success
	“privilege =	Success
	read =	Success
	representation =	Success

	sort =	Partial-Success
	view =	Success
Auditability:	authorship =	Success
	id =	Success
	worksheet =	Success
	finalize auditable =	Success
	summarize =	Success
	deletion =	Failure
	examine =	Success
Confidentiality:	Confidential =	Success
	private =	Success
	circumspection =	Success
	data protection =	Success
	information protection =	Success
Configuration:	architecture =	Success
	infrastucture =	Success
Accuracy:	accurate =	Success
	consistent =	Success
	time =	Success
	correctness =	Success
	exact =	Success
	standard =	Success
	"error free =	Success
Availability:	Availablity =	Success
	online =	Success
	integrity =	Success
	maintainence =	Success
	period =	Success
	resourse =	Success
Compliance:	compliance =	Success

	submission =	Partial-Success
	compliancy =	Success
	collabration =	Success
	teamwork =	Success
	prostration =	Success
	act in accord with accepted standard =	Success
	standards =	Success
	conform to requirements =	Success
Efficiency:	Efficient =	Success
	productive =	Success
	compatibility =	Success
	useful =	Success
documentation:	evidence =	Success
	testimonial =	Success
	authentication =	Success
	verfication =	Success
	process =	Success
	“describe =	Success
	define =	Success
	specify =	Partial-Success
	report information =	Success
Legal	legal =	Success
	lawful =	Success
	legitimate =	Success
	authorized =	Success
	permissible =	Success
	reuse =	Success
	vocabilary =	Success
Multiligual:	Multiligual Support =	Failure
	Icons Language =	Success

Performance:	interpretation =	Success
	Less memory required =	Success
	response time =	Success
	throughput =	Success
	simultaneous =	Success
	scale =	Success
Reliablity	sureness	Success
	database	Success
	input	Success
	validate	Success
Scalability	scalable	Success
	fast response	Success
	“mutliple devices at same time	Success
Security	Secure	Success
	password protected	Success
Interperability	quality	Success
	compatibility	Success
Usability	easy to use	Success
	Low perceived workload	Failure
	goals are easy to accomplish	Success
User Interface	User friendly	Success

Inability to recognize NFRs in key sentences is compared to difficulty in recognizing NFRs in other sentences. The mean of testing NFRs is shown on axis 1, the Y-axis on either sideshows the number of expressions not recognized by the NFR, and the right Y-axis shows how many expressions are not recognized by the NFR. The graph shows how many applications failed to validate NFR due to NFR heuristics. In the case of NERV and CEP methods, 3.51% and 1.75% of baseline NFRs, respectively, cannot be determined by heuristic method. According to the results of the analysis, the failure rate of NERV method and CEP method to detect the NFR line on the appropriate line is 1.76% and 0.0%, respectively. The number of errors to be detected in CEP andthe proposed method is the same as the previous findings, while the number of errors to be foundin the proposed method is the same as the previous findings. Form eProcurement has 66 requiredfields, which is too many. As noted in Section V-A, more than one type of NFR appears to appearin an imperative sentence depending on the context. NFR is defined as a sentence that is excludedor marked as such by the heuristic. The percentages of NFR articles are shown in the table below.This is important to consider, as there can be more than one NFR in a mandatory sentence. As a result, the failure detection (NFR) number of 57 messages required to be made in the success category, partial success and failure are based on the following: The total number of successful NFRs for the NFRElicit method is 92.04% compared to 81.82. Compared with 90.91% for the NERV method and the CEP method, it is 10.22% and 1.13% higher than the NERV method, respectively. In Figure 6, the path is represented by the X-axis and the number of successful NFRsin the desired sentence is represented by the left-hand Y-axis. The right Y-axis represents the ratio of NFRs. According to the results of the NORMAP index, there are 88 NFRs in the utility database. Based on the recognition criteria set out in Section V-A, more than one type of NFR may appear in a single sentence, depending on context. As shown in Table 6, specific information of some NFRs found by the heuristic method is given along with their percentages. â€‹gogo, part of the process has been completed and is not running. Also, comparing features or artifacts used in current and previous work will help clarify the heuristic as shown in the table below. Explanatory notes and values â€‹â€‹were used in all studies. The NFRElicit heuristic draws on initial work history, expert collaboration, and application context. The CEP method is the only method that can subtract and add NFR from the image. NFRElicit and CEP transactions etc. for automatic withdrawal. NERV and CEP methods are used to prioritize NFRs at fixed levels. NERV and NFRElicit technologies are the first to create a unique NFR card concept. Due to its widespread use, it is clear that the NFRElicist process includes not only the features of NERV or CEP, but also other features such as the content of the application, the role of the expert, the history of theproject. In figure 2 graph shows the performance of ML classifier for various NFRs of software.

NFR Identified by ML Classifier

NFRs	**Precision Percentage**	**Recall Percentage**	**Accuracy Percentage**
Accessibility:	77.00%	74.00%	75.00%
Auditability:	89.60%	92%	89%
Confidentiality:	70.00%	39%	0.50%
Configuration:	88.00%	47%	61%
Accuracy:	78.00%	65.00%	71%
Availability:	52.00%	50%	51%
Compliance:	64.00%	53%	58%
Efficiency:	82.00%	82.00%	82%
Documentation:	50.00%	17%	25%
Legal	57.00%	55%	56%
Multilingual:	71.00%	77%	74%
Performance:	66.00%	72%	69.00%
Reliability	76.00%	56%	81%
Scalability	52.00%	84%	56%

Security	45.00%	76%	43%
Interoperability	87.00%	45%	49%
Usability	91.00%	61%	21.00%
User Interface	61.00%	45%	89%

Table 2: Non-Functional Requirements Identified by Semi Supervised Approach

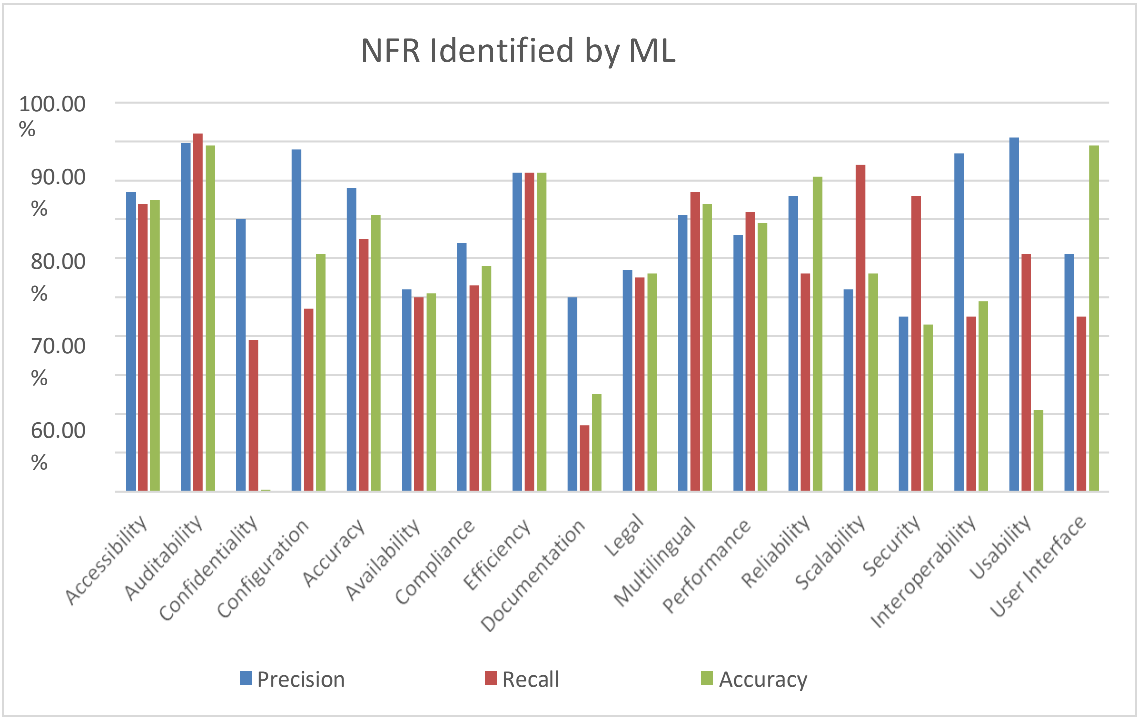

Figure 2: Graph showing the performance of ML classifier for identified NFRs

Conclusion and Future Work

In order to "choose" the software design, it is necessary to predetermine the non-functional requirements. After a software development project is designed, it should be implemented. Using traditional explanations translated into specifications is a popular technique for writing code for these levels. NFR recognition and classification is a complex problem and many text tracking methods have been proposed in the literature that have proven useful. When it comes to maintenance, it is necessary to leverage many of the predefined rules to separate units accurately enough to distinguish between new kinds of special requirements. For the classification of NFR, we developed and evaluated a semi-supervised study as described in this article, which aims to reduce the study by using elements such as word joins in materials of the same class. Reducing thepreviously mentioned learning needs through the use of text. A demand analysis support system incorporating this technology can help analysts save time by reducing the time spent on self- editing and analyzing already existing text. We aim to incorporate active learning into this reclassification process to reduce enrollments while maintaining accuracy. This will be done by choosing examples that analysts have written wisely as part of our ongoing research (for example, analysts should be asked to write more sample reports than best examples). Semi-supervised studies seem to be a promising method for determining NFRs based on experimental results. According to empirical evidence, semi-supervised algorithms require less effort than fully- controlled algorithms for tags and can be further developed as analysts if they are included in theneeds-controlling decision support system. Other benefits include making noise, not relying on thepast (e.g., past work) that could introduce noise into the text due to differences in vocabulary used by motivational groups. Unlike maintenance deployment, semi-maintenance deployment softwareno longer requires a "registration model".

References

[1] Casamayor, A., Godoy, D., & Campo, M. (2010). Identification of non-functional requirements in textual specifications: A semi-supervised learning approach. *Information and Software Technology*, *52*(4), 436-445.

[2] Casamayor, A., Godoy, D., & Campo, M. (2009). Semi-Supervised Classification of Non-Functional Requirements: An Empirical Analysis. *Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial*, *13*(44), 35-45.

[3] Tóth, L., &Vidács, L. (2018, May). Study of various classifiers for identification and classification of non-functional requirements. In *International Conference on Computational Science and Its Applications* (pp. 492-503). Springer, Cham.

[4] Younas, M., Jawawi, D. N., Ghani, I., & Shah, M. A. (2020). Extraction of non-functional requirement using semantic similarity distance. *Neural Computing and Applications*, *32*(11), 7383-7397.

[5] Binkhonain, M., & Zhao, L. (2019). A review of machine learning algorithms for identification and classification of non-functional requirements. *Expert Systems with Applications: X*, *1*, 100001.

[6] Shah, U. S., Patel, S., &Jinwala, D. (2016). Specification of Non-Functional Requirements: A Hybrid Approach. In *REFSQ Workshops*.

[7] Rashwan, A., Ormandjieva, O., & Witte, R. (2013, July). Ontology-based classification of non-functional requirements in software specifications: a new corpus and svm-based classifier. In *2013 IEEE 37th Annual Computer Software and Applications Conference* (pp. 381-386). IEEE.

[8] Tóth, L. (2018, June). Preliminary Concepts for Requirements Mining and Classification using Hidden Markov Model. In *THE 11TH CONFERENCE OF PHD STUDENTS IN COMPUTER SCIENCE* (p. 109).

[9] Younas, M., Wakil, K., Jawawi, D. N., Shah, M. A., & Mustafa, A. (2019). An automatedapproach for identification of non-functional requirements using Word2Vec model. *Int. J.Adv. Comput. Sci. Appl*, *10*(8), 539-547.

[10] Shah, U., Patel, S., &Jinwala, D. C. (2021). Detecting Intra-Conflicts in Non-Functional Requirements. *International Journal of Uncertainty, Fuzziness and Knowledge-BasedSystems*, *29*(03), 435-461.

[11] Kumar, M. S., & Harika, A. (2020). Extraction and classification of Non-Functional Requirements from Text Files: A Supervised Learning Approach. *Psychology and Education Journal*, *57*(9), 4120-4123.

[12] Sabir, M., Chrysoulas, C., &Banissi, E. (2020, April). Multi-label classifier to deal with misclassification in non-functional requirements. In *World Conference on Information Systems and Technologies* (pp. 486-493). Springer, Cham.

[13] Sabir, M., Chrysoulas, C., &Banissi, E. (2020, April). Multi-label classifier to deal with misclassification in non-functional requirements. In *World Conference on Information Systems and Technologies* (pp. 486-493). Springer, Cham.

[14] Quintanilla Portugal, R. L. Speeding-Up Non-Functional Requirements Elicitation.

[15] Naumcheva, M. (2021). Deep Learning Models in Software RequirementsEngineering. *arXiv preprint arXiv:2105.07771*.

[16] Dias Canedo, E., & Cordeiro Mendes, B. (2020). Software Requirements Classification Using Machine Learning Algorithms. *Entropy*, *22*(9), 1057.

[17] Devi, B. M., & Devi, M. S. (2021). AUTOMATIC CLASSIFICATION AND EXTRACTION OF NON-FUNCTIONAL REQUIREMENTS FROM TEXT FILES: A SUPERVISED LEARNING APPROACH. *European Journal of Molecular & Clinical Medicine*, *7*(9), 2231-2239.

[18] Talele, P., &Phalnikar, R. (2021, January). Classification and Prioritisation of Software Requirements using Machine Learning–A Systematic Review. In *2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence)* (pp. 912- 918). IEEE.

[19] Ahmad, A., Feng, C., Khan, M., Khan, A., Ullah, A., Nazir, S., & Tahir, A. (2020). A systematic literature review on using machine learning algorithms for software requirements identification on stack overflow. *Security and Communication Networks*, *2020*.

[20] Al Qaisi, H., Quba, G. Y., Althunibat, A., Abdallah, A., &Alzu'bi, S. (2021, July). An Intelligent Prototype for Requirements Validation Process Using Machine Learning Algorithms. In *2021 International Conference on Information Technology (ICIT)* (pp. 870-875). IEEE.

[21] Sun, P. (2021). A multi-layered desires based framework to detect evolving non-functionalrequirements of users (Doctoral dissertation, Iowa State University).

[22] Limaylla, M. I., Condori-Fernandez, N., &Luaces, M. R. (2021). Towards a Semi- Automated Data-Driven Requirements Prioritization Approach for Reducing Stakeholder Participation in SPL Development. *Engineering Proceedings*, *7*(1), 27.

[23] Nguyen, T. (2021). *Cross-applicability of ML classification methods intended for (non-) functional requirements* (Master's thesis, University of Twente).

[24] Cirqueira, D., Nedbal, D., Helfert, M., &Bezbradica, M. (2020, August). Scenario-based requirements elicitation for user-centric explainable ai. In *International Cross-Domain Conference for Machine Learning and Knowledge Extraction* (pp. 321-341). Springer,Cham.

[25] Deshpande, G., Motger, Q., Palomares, C., Kamra, I., Biesialska, K., Franch, X., ... & Ho, J. (2020, August). Requirements dependency extraction by integrating active learning with ontology-based retrieval. In *2020 IEEE 28th International Requirements Engineering Conference (RE)* (pp. 78-89). IEEE.

[26] Bernardi, S., Gentile, U., Nardone, R., &Marrone, S. (2020). Advancements in knowledge

elicitation for computer-based critical systems.

[27] Tahvili, S., Hatvani, L., Ramentol, E., Pimentel, R., Afzal, W., & Herrera, F. (2020). A novel methodology to classify test cases using natural language processing and imbalanced learning. *Engineering applications of artificial intelligence*, *95*, 103878.

[28] Nijanthan, L. (2021). *Improving the clarity of requirements by generating RCI value using Machine Learning* (Doctoral dissertation).

[29] Kravari, K., Antoniou, C., &Bassiliades, N. (2021). SENSE: A Flow-Down Semantics- Based Requirements Engineering Framework. *Algorithms*, *14*(10), 298.

[30] Lunarejo, M. I. L. (2021, September). Requirements prioritization based on multiple criteria using Artificial Intelligence techniques. In *2021 IEEE 29th International Requirements Engineering Conference (RE)* (pp. 480-485). IEEE.

[31] Henriksson, A., &Zdravkovic, J. (2021). Holistic data-driven requirements elicitation in the big data era. *Software and Systems Modeling*, 1-22.