Intelligent Chatbot for Requirements Elicitation and Classification

Chetan Surana Rajender Kumar Surana
Department of Computer Science and Engineering
Ramaiah University of Applied Sciences
Bengaluru, India
chetansurana015@gmail.com

Shriya
Department of Computer Science and Engineering
Ramaiah University of Applied Sciences
Bengaluru, India
21shriya@gmail.com

Dipesh B. Gupta
Department of Computer Science and Engineering
Ramaiah University of Applied Sciences
Bengaluru, India
dipesh4262@gmail.com

Sahana P. Shankar
Assistant Professor, Department of Computer Science and Engineering
Ramaiah University of Applied Sciences
Bengaluru, India
https://orcid.org/0000-0001-8977-9898

***Abstract*—Software Requirements (SR) are considered as the foundation for a supreme quality software development process and each step of the software development process is dependent and is related to the SR. Software requirements elicitation may be the most important area of requirements engineering and possibly of the entire software development process. There is a lot of human work required in the process of software requirements elicitation and software requirements classification and in cases where the requirements are huge in number, this requirements elicitation and classification process becomes tedious and is prone to errors. We propose a novel approach to automate Requirements Elicitation and Classification using an intelligent conversational chatbot. Utilizing Machine Learning and Artificial Intelligence, the chatbot converses with stakeholders in Natural Language and elicits formal system requirements from the interaction, and subsequently classifies the elicited requirements into Functional and Non-functional system requirements.**

Keywords—Requirements Engineering, Automation, Software Requirements Elicitation, Software Requirements Classification, Artificial Intelligence, Machine Learning, Chatbot, Classifier, Naïve Bayes, Support Vector Machine

I. INTRODUCTION

In the entire process of Software Development Life Cycle (SDLC), the requirements gathering and engineering phase plays a very important role [1]. It is the inception stage and the success at this stage has a great impact on the success of the later stages and the overall project [2]. Failure to gather complete, consistent and unambiguous system requirements can lead to complete failure of development of a software project [3]. The information at this stage is present in an unstructured format and this information is considered as a major source of input to the Software Architects in the Design Phase. Some of the issues that are encountered in the Requirements Engineering Phase includes the poor quality of requirements where the requirements are ambiguous, obsolete, inadequate, incomplete, inconsistent [4]. The quality of gathered requirements is also dependent on the merit and the experience of the Requirement Engineers and many a times due to factors such as inexperience or lack of domain familiarity of the Requirement Engineers, correct and complete set of requirements are not collected [5]. This manual method of requirement elicitation is also very tedious and is prone to errors especially in cases where the requirements are huge in numbers. Early detection and analysis of non-functional requirements is also gaining importance, so as to facilitate proper architecture choice. However, manual classification of requirements into functional and non-functional categories is a daunting task, especially on large projects with large number of requirements [6].

The dependency on the personnel can be eliminated by using a Chatbot. A chatbot is defined in the Oxford English Dictionary as “A computer program designed to simulate conversation with human users, especially over the Internet [7].” It is an AI chat agent. A Knowledge Base is constructed containing requirements pertaining to various domains like Banking, Retail, E-commerce, Healthcare, Finance, Entertainment. This Knowledge Base further contains two parts, Knowledge Representation and Inference Engine. The chatbot makes use of this Knowledge Base. The chatbot uses Natural Language Understanding, Natural Language Generation, Text Pre-processing and several machine learning algorithms to understand the user’s response, take further actions and respond accordingly.

Most of the requirements collected are present in an unstructured format and in a free flow English. The conversion of this unstructured format to an organized and more formal representation is dependent on the Knowledge Base and also on the intellect and experience of the Requirement Engineers. Replacing the Requirement Engineers with a Chatbot is a very good and effective idea and can eliminate the dependency on human personnel to get the task done. In addition to this, it is also seen that the same requirements is interpreted in different ways by different Requirement Engineers and this leads to conflict of interest if more number of Requirement Engineers are participating in the requirement elicitation task. The qualities of the requirements collected is also dependent on the emotional mood, health and temper of the engineer and gets affected by it. On the contrary, a Chatbot is very stable.

After the process of Requirement Elicitation by the Chatbot, a classifier is implemented using algorithms like

978-1-7281-0630-4/19/$31.00 ©2019 IEEE

Naïve Bayes or Support Vector Machine. The classifier model categorizes the elicited requirements into functional and non-functional requirements.

II. Background

As we are seeking towards natural ways to integrate automation into everyday life, the conversational system in today's world has become prevalent as a basis for the computer human interaction. Some of the well-known examples of conversational Artificial Intelligence include Amazon's Alexa, Apple's Siri and Microsoft Cortana.

A. Natural Language Understanding and Natural Language Generation

NLU is said to be a sub field of Natural Language Processing(NLP) which converts the unstructured inputs into a structured one which can be understood by the machine and can be acted upon [8]. Natural Language Generation is defined as a subdivision of AI and it aims to cut down the communication gaps between humans and machines. This technology takes input in the non-linguistic form and converts it into a format which is understood by humans like documents, reports or texts [8]. Here, NLU is used to take input as a sentence or a statement and provide the intent and entities and a confidence score which is used by the chatbot. An intent tells us what the user wishes to do entities are defined as the attributes which provides description about the user's task. Intent Classification can be defined as a process in which a software function which is called as "classifier" identifies the input and then associates the information provided with a specific content thereby giving a detailed information of the words which could be understood by the computer. Some of the algorithms used for the intent classification are Pattern matching, Machine learning algorithms and neural networks [9]. Entity extraction can be defined as a process of identification and classification of key elements from the textual format into pre-defined categories and it helps in the transformation of the unstructured data to the data that in structured format, thereby making it machine readable and available for standard processing that can be used for purposes like retrieval of information, extraction of facts and answering of the questions. Entity extraction which is based on semantic technology includes entity relation extraction, linking and fact extraction [10].

B. Chatbot Dialog Management

The description of flow inside a conversation is described by the dialogue management. Dialogue management can be defined as finite state machines are dependent on acquired information and the messages received from the interlocutors. The main component that is used for the Chat Dialogue Management model is a recurrent neural network - a Long Short-Term Memory (LSTM) network. LSTM networks contain some internal contextual state cells and they act as long-term or short term memory cells. A representation of dialogue history which provides relief to the system developer from much of the manual feature engineering of dialogue state is automatically inferred by the LSTM. The aim of LSTM is taking actions in the real world on behalf of the user and these actions can be optimized with the help of supervised learning, where the example dialogues are provided by a domain expert which should be imitated by the LSTM [11].

C. Classification of Requirements

Text classification is used for automatic assignment of predefined categories to free-text documents. The aim of text classification is to provide conceptual organization to a large collection of documents and this can be done using classification algorithms like Support Vector Machine (SVM) or Multinomial Naïve Bayes (Multinomial NB) [12]. An SVM is a discriminative classifier which is formally defined by a separating hyperplane. Multinomial Naïve Bayes algorithm is a classification algorithm for binary (two-class) and multi-class classification problems [12].

III. Related Work

Calazans et al. in [13] highlight that Software Requirements Engineering activities such as elicitation, classification, analysis and specification are mostly carried out people, unlike other automated Software Engineering areas like programming and testing. They identify several competences categorized as Knowledge, Skills and Attitudes, stressing the importance of the role a requirements analyst. In [14], Meth et al. underscore that identification of user requirements is an important and crucial activity, with several efforts devoted to utilize tools and technologies to enhance requirements elicitation activity. Of the thirty-six papers they analyzed, twelve were devoted to requirements model generation and eleven were concerned with Requirements Identification. In [15], Harman discusses approaches towards application of Artificial Intelligence in Software Engineering and potential challenges. "The SE community has used three broad areas of AI techniques: 1) Computational search and optimization techniques (the field known as Search Based Software Engineering (SBSE)), 2) Fuzzy and probabilistic methods for reasoning in presence of uncertainty, 3) Classification, learning and prediction." [15] In [16], Friesen et al. propose CORDULA (Compensation Of Requirements Descriptions Using Linguistic Analysis), "a system using chatbot technology to establish end-user communication in order to support the requirement elicitation and partial compensation of deficits in user requirements." In [5], Abad et al. propose an automated tool to assist analysts in eliciting requirements from stakeholders about projects from an unfamiliar domain. In [17], Aysolmaz et al. propose a semi-automated approach to generate requirements from business process models. In [18], Mahmoud and Williams propose an "unsupervised approach for detecting, classifying, and tracing non-functional software requirements (NFRs). The proposed approach exploits the textual semantics of software functional requirements (FRs) to infer potential quality constraints enforced in the system." In [19], Navarro-Almanza et al. discuss application of Deep Learning "to classify software requirements without labor intensive feature engineering." Their proposed model is based on Convolutional Neural Network (CNN). Their model was evaluated using PROMISE corpus.

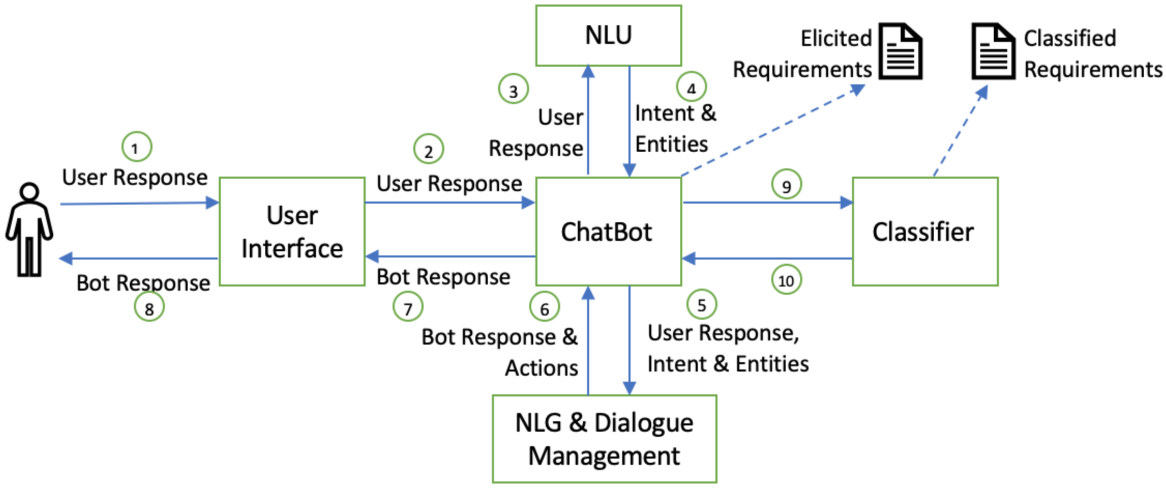

Fig. 1 Block Diagram Representing How the System Elicits and Classifies the Requirements

IV. Proposed Work

A. *Algorithm*

Start

1. Input user's response
2. Preprocess the user response text
3. Determine the intent from the user response using intent classification algorithm
4. Extract relevant entity values from the user response using entity extraction algorithm
5. Using a dialogue management model, predict the next action and bot response based on context and present user intent. The actions appropriately elicit and write a system requirement to file from the user response
6. Repeat 1 to 5 until user intent is to end the conversation
7. Classify the elicited requirements using a text classification algorithm
8. Display elicited and classified requirements to user and save them to file as deliverables.

End

An automated approach to elicit formal system requirements is proposed using a conversational chatbot. The elicited requirements are categorized into functional and non-functional requirements using a classification algorithm. The proposed work consists of developing the user interface, the chatbot and a text classifier model using Python. Fig. 1 is a block diagram depicting the essential components of the proposed system. The user interface has been developed with the aid of PyQt5 framework. The user interface presents the chat window where the user can input his response using a text field or audio. The chat window displays the bot response and the history of conversation between the user and the chatbot. When the user responds to end the conversation, the user interface changes screens to display the formal system requirements elicited and classified into functional and non-functional requirements. The user interface passes the user response to the chatbot model and receives the bot response.

The chatbot uses Natural Language Understanding, Natural Language Generation, Text Pre-processing and several machine learning algorithms to understand the user's response, take further actions and respond accordingly. Rasa-NLU and Rasa-Core opensource frameworks are utilized for assisting in Natural Language Processing and Dialogue Management. A training data file is prepared that consists of various possible intents and example user responses for each intent, as shown in Fig. 2. The entity and values are also specified in the examples. Using this training data, a Rasa NLU model for intent classification and entity extraction model is trained. This model classifies the user response to the most probable intent label and gives the values of identified entities. Word embeddings are used as vector representation of the words, expressing their semantic and syntactic meaning. A support vector classifier is then trained on these word embeddings to classify them to most likely intent. The entity extraction is done using Conditional Random Field, which predicts labels for sequences of words. Python's Sklearn machine learning library is used for the classification, and spaCy English language model is used for Named Entity Recognition.

```
## intent: what_are_you
- what are you?
- what can you do?
- what are you capable of?
- what are you used for?

## intent:open_account_by_type
- I want to be able to create a [savings](ACCOUNT_TYPE) account
- the user should be able to open a new [current](ACCOUNT_TYPE) account
- the system should provide a feature to open a [savings and current](ACCOUNT_TYPE) account
- I should be able to create a [savings](ACCOUNT_TYPE) account
```

Fig. 2 Training Data Snippet for Intent Classification and Entity Extraction

For the dialogue management, a training data file is prepared that consists of several sample user and bot conversations, as shown in Fig. 3. In the conversation, the user response is represented by its intent, and the bot response is represented by the bot's utterance and actions taken. A Rasa Core model is trained with this data, which develops and trains an LSTM Recurrent Neural Network (RNN). The LSTM

```
## greeting
* greeting
- utter_greeting

## open account
* open_account
- utter_ask_account_type
* account_type
- elicit_open_account
- utter_what_else
```

Fig. 3 Training Data Snippet for Dialogue Management Model

```
Requirement, Category
The system should maintain order status,Functional
The sytsem should track location and update ETA,Functional
The system should allow the user to book rooms,Functional
The system should be highly secured and unable to be hacked,Non-functional
The system should be reliable enough to sustain in any condition,Non-functional
The system should multi language support,Non-functional
```

Fig. 4 Requirements Dataset Snippet used by Classification Model

automatically infers dialogue history and maps the raw dialogue history to a distribution over system actions. The system actions can be specified as text that the bot utters in reply to the user, or as code that is to be executed. The chatbot is trained with data so as to identify system requirements from the user response, extract entity values, ask for missing information from the user if necessary, write the requirement to file, and converse with the user in natural language. The system requirements gathered from the user interaction are written to a file.

For classifying the elicited requirements into Functional and Non-functional categories, a text classification model is developed. A training data set, shown in Fig. 4, was prepared with over eight hundred samples labelled as Functional and Non-functional, describing system requirements from across domains, and collected from the Internet and books. The input samples are represented as feature vectors using Bag-of-Words approach and tf-idf frequency. Using these feature vectors, text classification models are developed using Multinomial NB and SVM algorithms.

V. RESULTS

Fig.5 shows an example interaction between the user and the chatbot, which begins with the user and the chatbot greeting each other. The chatbot has been trained to accept and gather requirements for a Net-banking app system. The user describes the system features and the chatbot extracts relevant and complete system requirements from the interaction. When the user specifies an incomplete functional requirement to open an account, the chatbot identifies that account type data is missing and retrieves it from the user through a subsequent question. The user also specifies that the system should be portable, a Non-functional requirement. The chatbot is able to clearly identify the intent of the user, whether it is simple conversation or important statements where a system requirement is being described. The chatbot can capture requirements described in a single sentence, or through several questions, depending on completeness of user response. Fig.6 shows the formal system requirements elicited from the sample interaction, and Fig.7 shows the requirements as correctly classified into Functional and Non-functional categories.

Fig. 8 shows a comparison of the accuracy of the two text classification algorithms - Multinomial NB and SVM, on the

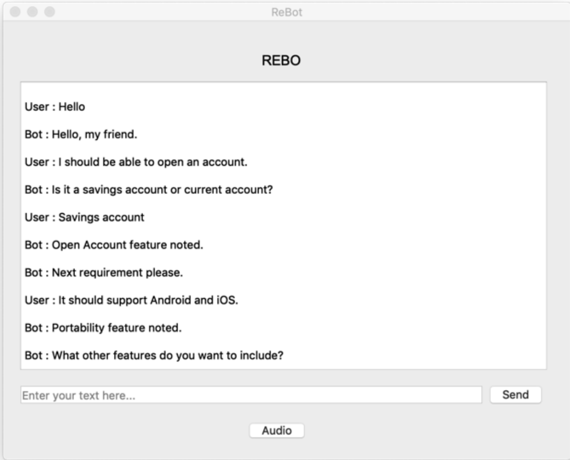

Fig. 5 Sample User and Chatbot Interaction

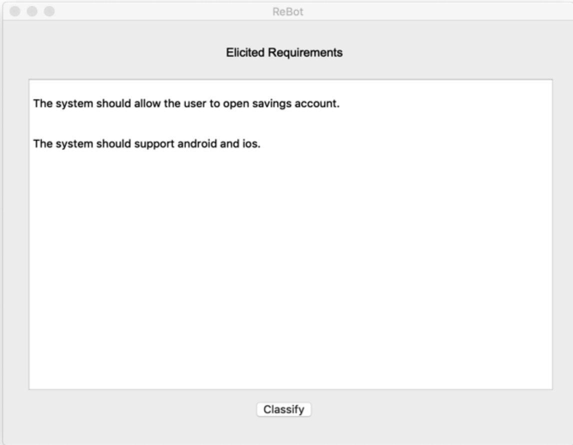

Fig. 6 Elicited Formal System Requirements

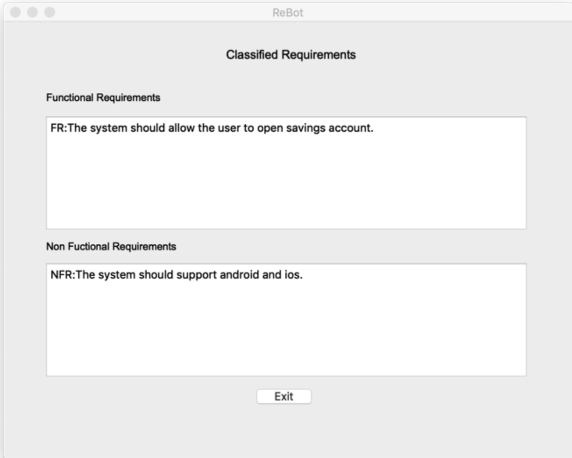

Fig. 7 Classified System Requirements

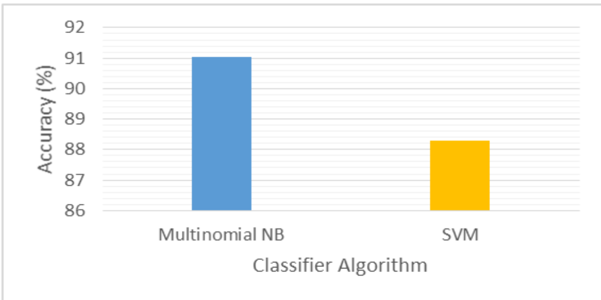

Fig. 8 Comparison of Accuracy using Multinomial NB and SVM Classification Algorithms

requirements dataset. Table 1 lists the accuracy, precision, recall and F1-score performance measures of the two algorithms. Both algorithms yield comparably good performance. As the requirements dataset is small, and the samples are snippets, Multinomial Naïve Bayes algorithm performs slightly better and is conveniently sufficient for the classification of requirements into Functional and Non-functional categories.

Table I Performance Evaluation of the Two Classification Algorithms on the Collected Requirements Dataset

Algorithm	**Performance Parameter**			
	Accuracy	***Precision***	***Recall***	***F1-Score***
Multinomial NB	0.91	0.91	0.91	0.91
SVM	0.88	0.88	0.88	0.88

VI. CONCLUSION

A chatbot is one of the simplest ways to transfer data from a computer without hustling and thinking much about the proper keywords to look up in a search or browse several web pages to collect information. The queries can be easily typed by the users in the natural language and retrieve the information. In this paper, automation of requirements elicitation and classification using conversational chatbot technology and text classification algorithms has been presented. A Chatbot is a great tool for quick interaction with the user. With intelligent models, it can control the dialogue flow, extract essential details from the user's responses and present vital information. The work presented here serves to portray that requirements elicitation can be automated using an artificially intelligent chatbot. The elicited requirements can then be classified into Functional and Non-functional. Identification of a Non-functional requirement and the category aids in identifying required architectural design for fulfilling that requirements. The proposed work carried out, however, has several limitations. The chatbot developed is trained to capture a limited set of requirements from a single domain. The chatbot should be improved with extensive and intelligently crafted training data so as to be able to capture a complete set of system requirements. There is no existing dataset of stakeholder and requirement analyst conversations and the resulting system requirements elicited. Therefore, such a dataset needs to be prepared to assess the ability of the chatbot to elicit a system's complete set of requirements. Presently, the chatbot can identify incomplete description of a system requirement, and ask further questions to gather complete information. Further work and investigation is to be carried out to elicit unambiguous requirements from the stakeholder's responses. The classification performance of the system requirements is significant. It can be further improved by fine tuning the algorithm parameters, and also testing with deep learning models. The system can be extended to carry out feasibility prediction, where it predicts which system requirements can be implemented and which are not feasible. A model can be developed to estimate the likely cost, time and man-hours required for project completion.

REFERENCES

[1] Atas, M., Samer, R. and Felfernig, A., 2018, December. Automated Identification of Type-Specific Dependencies between Requirements. In 2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI) (pp. 688-695). IEEE.

[2] Chen, F., Power, N., Collins, J.J. and Ishikawa, F., 2019, April. Contemporary requirements challenges and issues: an empirical study in 11 organizations. In Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing (pp. 1592-1599). ACM.

[3] Shafiq, M., Zhang, Q., Akbar, M.A., Khan, A.A., Hussain, S., Amin, F.E., Khan, A. and Soofi, A.A., 2018. Effect of project management in requirements engineering and requirements change management processes for global software development. IEEE Access, 6, pp.25747-25763.

[4] Ferrari, A., Spoletini, P. and Gnesi, S., 2016. Ambiguity and tacit knowledge in requirements elicitation interviews. Requirements Engineering, 21(3), pp.333-355.

[5] Abad, Z.S.H., Gervasi, V., Zowghi, D. and Barker, K., 2018, August. ELICA: An Automated Tool for Dynamic Extraction of Requirements Relevant Information. In 2018 5th International Workshop on Artificial Intelligence for Requirements Engineering (AIRE) (pp. 8-14). IEEE.

[6] Jan Khan Roohullah et al. "Survey : Dealing Non-Functional Requirements At Architecture Level".Transactions on Software Engineering 9.2 (2016),pp. 7–13.

[7] The Oxford English dictionary. (2004). Oxford: Oxford University Press.

[8] Gatt, A. and Krahmer, E., 2018. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. Journal of Artificial Intelligence Research, 61, pp.65-170.

[9] Rahman, A.M., Al Mamun, A. and Islam, A., 2017, December. Programming challenges of chatbot: Current and future prospective. In 2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC) (pp. 75-78). IEEE.

[10] Khan, R. and Das, A., 2018. Build Better Chatbots. Apress.

[11] Wen, T.H., Gasic, M., Mrksic, N., Su, P.H., Vandyke, D. and Young, S., 2015. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. arXiv preprint arXiv:1508.01745.

[12] Setyawan, M.Y.H., Awangga, R.M. and Efendi, S.R., 2018, October. Comparison Of Multinomial Naive Bayes Algorithm And Logistic Regression For Intent Classification In Chatbot. In 2018 International Conference on Applied Engineering (ICAE) (pp. 1-5). IEEE.

[13] Calazans, A.T.S., Paldes, R.A., Masson, E.T.S., Brito, I.S., Rezende, K.F., Braosi, E. and Pereira, N.I., 2017, September. Software requirements analyst profile: A descriptive study of brazil and mexico. In 2017 IEEE 25th International Requirements Engineering Conference (RE) (pp. 204-212). IEEE.

[14] Meth, H., Brhel, M. and Maedche, A., 2013. The state of the art in automated requirements elicitation. Information and Software Technology, 55(10), pp.1695-1709.

[15] Harman, M., 2012, June. The role of artificial intelligence in software engineering. In 2012 First International Workshop on Realizing AI Synergies in Software Engineering (RAISE) (pp. 1-6). IEEE.

[16] Friesen, E., Bäumer, F.S. and Geierhos, M., 2018. CORDULA: Software Requirements Extraction Utilizing Chatbot as Communication Interface. In REFSQ Workshops.

[17] Aysolmaz, B., Leopold, H., Reijers, H.A. and Demirörs, O., 2018. A semi-automated approach for generating natural language requirements documents based on business process models. Information and Software Technology, 93, pp.14-29.

[18] Mahmoud, A. and Williams, G., 2016. Detecting, classifying, and tracing non-functional software requirements. Requirements Engineering, 21(3), pp.357-381.

[19] Navarro-Almanza, R., Juárez-Ramírez, R. and Licea, G. (2017). Towards Supporting Software Engineering Using Deep Learning: A Case of Software Requirements Classification. 5th International Conference in Software Engineering Research and Innovation (CONISOFT)