

Use of Natural Language Processing in Software Requirements Prioritization – A Systematic Literature Review

Suchetha Vijayakumar ^{1,2} & Nethravathi P. S. ³

¹ Research Scholar, College of Computer and Information Sciences, Srinivas University, Mangalore

² AIMIT, St. Aloysius College (Autonomous), Mangalore

OrcID : 0000-0001-8379-5651; Email : such_vijay@yahoo.com

³ Professor, College of Computer and Information Sciences, Srinivas University, Mangalore

OrcID : 0000-0001-5447-8673; Email : nethrakumar590@gmail.com

Subject Area: Computer Science.

Type of the Paper: Review based Research Agenda.

Type of Review: Peer Reviewed as per [C/O/P/E](#) guidance.

Indexed In: OpenAIRE.

DOI:

Google Scholar Citation: [IJAEML](#)

How to Cite this Paper:

Suchetha, V. & Nethravathi, P. S., (2021). Use of Natural Language Processing in Software Requirements Prioritization – A Systematic Literature Review. *International Journal of Applied Engineering and Management Letters (IJAEML)*, 5(2), 152-174. DOI:

International Journal of Applied Engineering and Management Letters (IJAEML)

A Refereed International Journal of Srinivas University, India.

Crossref DOI : <https://doi.org/10.47992/IJAEML.2581.7000.0110>

© With Authors.



This work is licensed under a [Creative Commons Attribution-Non-Commercial 4.0 International License](#) subject to proper citation to the publication source of the work.

Disclaimer: The scholarly papers as reviewed and published by the Srinivas Publications (S.P.), India are the views and opinions of their respective authors and are not the views or opinions of the S.P. The S.P. disclaims of any harm or loss caused due to the published content to any party.

Use of Natural Language Processing in Software Requirements Prioritization – A Systematic Literature Review

Suchetha Vijayakumar ^{1,2} & Nethravathi P. S. ³

¹ Research Scholar, College of Computer and Information Sciences, Srinivas University,
Mangalore

² AIMIT, St. Aloysius College (Autonomous), Mangalore

OrcID : 0000-0001-8379-5651; Email : such_vijay@yahoo.com

³ Professor, College of Computer and Information Sciences, Srinivas University, Mangalore

OrcID : 0000-0001-5447-8673; Email : nethrakumar590@gmail.com

ABSTRACT

Purpose: *Research involves the creation and implementation of new ideas by keeping the existing work as a foundation. The literature review done in this paper is to familiarise and to know about the domain of research to integrate the existing ideas with the new ones.*

Methodology: *The literature that is required for this study is chosen from multiple secondary sources such as journals, conference proceedings, and web resources. All the pieces of literature are carefully studied and summarised. This is further used to arrive at Research agendas and Research gaps.*

Findings/Result: *It has been observed and understood that Natural Language Processing (NLP) is a field involving analysis and processing of textual contents. It also requires Machine Learning Algorithms to support the processing. This combination has already been used in various domains, the important one being the health sector. EMR data is huge and NLP can successfully process and prioritize them in different dimensions. In that direction, the same concept and technology can be applied to Software Engineering also and Requirements can be prioritized.*

Originality: *This literature review study is carried out using secondary data which is collected through various online sources. The information thus gathered will be used in the future to build upon existing theory and framework or build a new methodology. It is also seen that any conclusion or decision is not biased or unidirectional. A sincere effort is made to identify a research topic to carry out the research.*

Paper Type: *Literature Review.*

Keywords: Software Development, Software Requirement Prioritization, Natural Language Processing, Machine Learning, Open Source NLP tools,

1. INTRODUCTION :

Software Development is a multistep, comparatively lengthy process where every stage of the development process includes milestones and deliverables. Requirement Elicitation (RE) of Requirement Gathering is one such stage that is considered as the heart of Requirements Engineering as it is highly critical, error-prone, and communication-dependent. Requirements Engineering is considered as an important and prominent subset of conventional Software Engineering or Development process as it involves using many tools and techniques to process the requirements and also sometimes developing new tools and techniques for the purpose [1]. RE phase in Software Development Life Cycle is meant to collect Requirements. These requirements are then classified as functional, non-functional, and domain specific requirements. These specifications are made available through the Software Requirement Specification (SRS) document. Functional requirements specify the functionalities that are expected out of the system to be developed and non-functional requirements are the features that affect the functionalities [2]. RE results in a huge amount of data and requirements from clients and stakeholders which is beyond the capability of humans to handle them and queue them in an orderly manner for implementation and consideration. The requirements thus elicited and classified are bound

to have lots of ambiguity. These ambiguities may be one among lexical ambiguity, semantic ambiguity, or pragmatic ambiguity. Though the reasons for having ambiguity in requirements are many, classification and prioritization are huge concerns [3]. Software Requirements Prioritization (SRP) comes into use at this stage of Requirements Engineering. SRP is required to specify the order in which the long list of requirements given by the client needs to be processed. Hence SRP deals with the order of processing requirements. SRP aids in a better way of scheduling requirements [4]. All requirements collected come under either of one of the categories as mentioned earlier. Further, Requirements can be classified as low priority, medium priority, and high priority based on which they can be prioritized. As different stakeholders will have various opinions regarding the list of requirements, it becomes invariably important to set up an order in which requirements need to be implemented. It is also necessary to prioritize requirements as one needs to keep track of cost, time, and resources too [5]. It is believed that more than 85% of the defects and bugs in software are because of incorrect requirements and also not considering the important requirements. Hence, SRP is an important step in the Software Development process which requires at most potential and knowledge [6].

Several SRP techniques have been formulated so far which take up the prioritization process in various dimensions. Every method has its own merits, demerits, and constraints. One single method is not suitable for all types of development processes. Some of them are suitable for large applications while some techniques work well when the requirements size is small. Accordingly, the prioritization techniques are classified [7]. Prioritizing Software Requirements are essential from a developer's perspective also. There will be requirements that are interdependent and also some of them would be implemented parallelly. Hence an appropriate technique is necessary to prioritize Functional Requirements to make developers work easy [4]. Moreover, Prioritizing Software requirements is an important activity as it brings down the problems and issues associated with resources, manpower, budget, etc. Not only this, the Prioritization process helps in getting rid of unwanted requirements which otherwise would cost a lot for the development team.

Requirements are gathered at the beginning of the software development process. These are then prioritized according to their relative importance to the market and the product itself, considering several factors that can influence prioritization decisions [8]. Hence, Requirements Prioritization is a process that is just a step after the classification of requirements. This process involves three steps – first one is to find out which of the requirements need to be included and which of them need to be ignored; the second step is to find out if there are any dependent requirements; and finally, the third step is to do a prioritization [9].

The following figure throws more light on Requirement Prioritization in the Requirements Elicitation process.

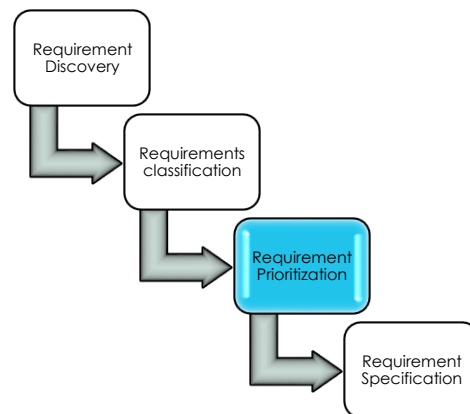


Fig. 1: Requirement Elicitation Process and the place of Requirement Prioritization

In this paper, a review of various Software Requirements Prioritization methods is done and also various avenues, possibilities, and technologies that can support the automatic prioritization process. The paper is organized as follows – the next section talks about the Research objective. This is followed by a literature survey done on the topic and related topics. A total of about ten papers are reviewed and their findings are recorded and presented. There is a separate review done on the role of NLP in Software Engineering followed by a review on the use of ML and AI in NLP. The next section provides an

overview of various open source tools in NLP that are available for the purpose. This is followed by a small write up on repositories that are publicly available and contain Requirement Documents. A tabular review of Research papers in the domain and related work is also presented. A total of 50 papers are reviewed. The next section gives a brief overview of present work and possible future work. Research Gaps are identified in the next section which is also analyzed. In the end, a SLOC analysis is done on the topic.

2. RESEARCH OBJECTIVE AND METHODOLOGY :

Requirement Prioritization is considered to be the most significant step in Requirement Engineering which again comes under the umbrella of the Requirement Analysis phase of SDLC. Many methods and methodologies have been devised to prioritize requirements to make this step fruitful and beneficial and helps in analyzing the requirements of clients in a better way. Automating Requirements Prioritization is something that is the need of the hour as manual prioritization might be time consuming and biased to some extent. Finding ways of using technology, especially Natural Language Processing to do the task is the prime objective of this study. The other objectives include-

- Exploring different Requirement Prioritization techniques available
- Finding the various tools that are available to do Requirement classification
- Deciding a suitable technology for automated Requirement prioritization
- When multiple technologies are available for the purpose, finding the more appropriate ones
- Finding the right method/algorithm(s) to achieve Requirement Prioritization

3. LITERATURE SURVEY :

Olaronke, I., *et al* (2018) [10] have examined different techniques for prioritizing needs by stating their strengths and weaknesses. In the research paper, they have analyzed and compared technologies such as binary search tree, AHP, AHP hierarchical structure, priority group/numerical analysis, bubble classification, MoSCoW, simple classification, and planning game. It is also concluded that the numerical value assignment and simple classification methods require less time in the prioritization process, and their scalability and reliability are also lower. It has also been said that the AHP process takes more time to prioritize requirements; it is reliable but not scalable. The research study made by them has also shown that when multiple stakeholders are involved, it is difficult to prioritize requirements using existing prioritization techniques due to the different perspectives of different stakeholders.

Perini, A., *et al* (2013) [11] describes a novel requirements prioritization method called Case-Based Ranking (CBRank). CBRank combines stakeholders' preferences with requirements ordering approximations which are computed through Machine Learning techniques. It follows a method called case based approach in which solutions to a problem are derived through previously solved similar problems. Hence in Requirements Prioritization, previously ranked prioritizations are taken and approximate ranks are calculated for the new set of requirements. An algorithm called the RankBoost algorithm is also considered here for Prioritizing requirements.

In their paper, Ejaz, K., *et al* [12] have proposed a simple yet novel model for Requirements Prioritization. According to them, the requirements are given by the clients to the Project Manager who in turn compiles them into optimal data. This data helps in determining the cost and benefits which can tell about the feasibility of implementation. This is done with the help of a high level algorithm called the Greedy algorithm and a low level algorithm. The authors further state that this can be developed into an intelligent system with the help of technical attributes.

Choudhary S., *et al* [13], in their work have discussed about email prioritization using Machine Learning algorithms. Naïve Baye's Classifier is used here for the classification of emails. Here weights are assigned to certain words in emails. Then these weights are totaled up to prioritize emails based on weights. NLP is used to find important words and their corresponding meaning.

According to Singh, M., *et al* [14], verification and validation of radiology reports and their prioritization can be done by using a Naive Bayesian classifier. One hundred reports were used to train the model as high priority and low priority following which this model was used to evaluate three

hundred and fifty four reports. The model developed for this purpose using the Naïve Bayes classifier is also reported to have given an accuracy of 93.50%.

Amora, P. R. P., et al [15] in their research have combined sentiment analysis and prioritization of user comments and reviews. Tweets from social media by the customers are analyzed and classified to do a sentiment analysis. The tweets are then attended based on the priority so that the company will not lose any customers because of not attending to their dissatisfaction. Hence all comments and reviews are prioritized and actions are taken accordingly. To do this, the authors have used many Machine learning algorithms including a couple of deep learning algorithms such as LSTM and Bi-directional LSTM to achieve customer service prioritization.

In their paper, Muqem, M., *et al* [16] propose a fuzzy based approach for Requirement prioritization. Here the requirements prioritization is done immediately after requirement elicitation. Here the attributes are taken as fuzzy variables. Later these variables are processed in an inference system and the ranking is given by the expected value of the fuzzy variable.

Sharma, N., et al [17] have discussed about the direction of research in Software Engineering and Natural Language Processing. They have given a detailed list of Software Engineering domains according to SEBOK (Software Engineering Body of Knowledge). Furthermore, two approaches of NLP namely the Classical approach and Empirical & Statistical approach are specified. The authors also describe the methods to generate research questions and produce Research products in Software Engineering and NLP.

Vibha G., et al [18] have evaluated various software requirements prioritization methods available considering multiple attributes and also tried to identify ways to decide the prioritization method most suitable for an application. To do this, the authors have used Fuzzy Multi-Criteria Decision-Making Method (FMCDM) to choose an appropriate prioritization method. They have used an experimental study approach for finding out an appropriate method. Moreover, the newly proposed method is based upon the concept of fuzzy set theory.

In their paper Salado, A., et al [19] have spoken about deciding conflicting requirements. In this regard, the authors have proposed a new and elegant ARP model which provides information about priorities to use proper criteria on any decision at a given time. Also, this model helps handle priorities in a structured manner and aids in offering flexibility for specific needs of the project.

3.1 The role of NLP in Software Engineering:

Natural Language Processing (NLP) and Software Engineering – both domains have one feature in common that they are two important branches of Computer Science. The contributions of NLP and SE are enormous in their domains. Lots of research has been conducted to find various avenues of using NLP in SE. Lots of textual documents are generated during the Software Engineering process. Every phase of the Software Development Life Cycle has some artifacts generated at the end of every phase. A Software Requirement Specification Document is generated at the end of the Requirement Elicitation phase. Same way a Design document is generated at the end of the Design phase and various Test cases are documented during the testing phase. Hence, we see a wide range of opportunities to use NLP as the textual content is more [20].

In the Requirement Engineering phase itself, NLP can be applied in six different categories. They are Requirement classification, Requirement Prioritization, Requirement Ambiguity removal, Requirement Quality Assessment, and Analysis. The various techniques of NLP that can be applied for the above purpose are Tokenization, POS tagger, Text chunking, Parsing, VSM, TF-IDF [21]. UML class diagrams can be generated from the Requirements Specification document with the help of NLP and by applying other heuristic rules. NLP tools are devised which can map the knowledge extracted into UML class diagrams [22].

3.2 Use of Machine Learning and Artificial Intelligence in NLP:

NLP is a sub branch or field of Computer Science and computational linguistics which is driven by all the advances and researches in Machine Language (ML). NLP and ML are a part of Artificial

Intelligence (AI). AI is the umbrella under which NLP and ML exist. The following figure shows this relationship.

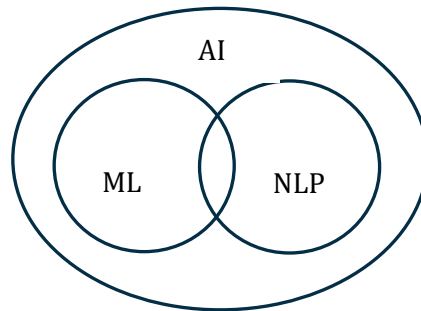


Fig 2: Relationship between AI, ML, and NLP

NLP is one of the text mining methods used for text analysis that implements a unique type of linguistic analysis that essentially makes the machine read and analyze [23]. ML and NLP together make a machine behave intelligently. Due to the advancement in NLP, the systems can accept information in a better way. AI helps systems to process the inputs and do better analysis and predictions by using Machine Learning algorithms and techniques [24].

Table 1: ML algorithm used for different NLP tasks

| NLP task | ML algorithm |
|---------------------------|--|
| Tokenization | Support Vector Machine Recurrent Neural Network |
| Stemming or Lemmatization | Random Forest Decision Tree |
| Syntactic Analysis | Random Forest Recurrent Neural Network K-Nearest Neighbour |
| Semantic Analysis | Random Forest Gradient Booster Decision Tree |

Information Source: Role of Machine Learning in Natural Language Processing by Dr. Tatwadarshi P. N. available at www.analyticsvidhya.com

3.3 Open Source NLP Tools:

In recent years, open source NLP tools have become more accessible. Unlike a decade ago when it was difficult to access tools for NLP, it is now the era of open source NLP tools. There are a huge number of tools available to do an automatic analysis of Natural language data. There are also separate tools developed for various languages too, while some of them can be used to analyze data of many languages. The following table summarises some of the open source NLP tools available.

Table 2: List of various Open Source NLP tools

| Name of the tool | Description | Tasks performed |
|-----------------------|--|--|
| Python NLTK [25] | It is a Python package with a bundle of libraries. | <ul style="list-style-type: none"> • Classification • Tokenization • Stemming • Tagging • Parsing • Semantic reasoning |
| Stanford CoreNLP [26] | CoreNLP is a Java based natural language processing tool supporting languages such | <ul style="list-style-type: none"> • Linguistic annotation for text • Boundaries for tokens and sentences |

| | | |
|---------------------|--|---|
| | as Arabic, Chinese, English, French, German, and Spanish. | <ul style="list-style-type: none"> • POS • Named entities • Coreference • Sentiment • Quote attributions |
| Apache OpenNLP [27] | Apache OpenNLP library is a machine learning based toolkit that is used to process natural language text. | <ul style="list-style-type: none"> • language detection • tokenization • sentence segmentation • POS tagging • named entity extraction • chunking • parsing • coreference resolution |
| MALLET [28] | MALLET is a Java-based package for statistical natural language processing | <ul style="list-style-type: none"> • document classification • clustering • topic modeling • information extraction • sequence tagging |
| GATE [29] | GATE is an NLP tool capable of solving any type of text processing problem. | <ul style="list-style-type: none"> • Tokenization • Gazetteer • sentence splitter • part of speech tagger • named entities transducer • co reference tagger |
| spaCy [30] | spaCy is an NLP tool mainly useful for advanced natural language processing written in Python and Cython. | <ul style="list-style-type: none"> • Tokenization • Lemmatization • Part-of-speech(POS) tagging • Entity recognition • Dependency parsing • Sentence recognition • Word-to-vector transformations • other cleaning and normalization text methods |
| Gensim [31] | Gensim is used for unsupervised topic modeling and natural language processing, using modern statistical machine learning implemented in Python and Cython | <ul style="list-style-type: none"> • Topic modeling |

Thus, it is seen that there are many tools available for analyzing Natural language. Some of these tools are written in Python, while tools like CoreNLP were originally written in Java. Unlike the scenario in the early 90s, many flavors of tools are available for Natural Language processing and analysis now. It is up to the users to decide the appropriate tool based on the need and necessities.

3.4 Repositories of SRS:

A Software Requirement Specification (SRS) document is the outcome of the Requirement Elicitation phase. This becomes the base for further developmental work that takes place. To carry out the automated Requirement Prioritization task, there must be standard SRS so that textual data from that can be analyzed and applied on algorithms being tested. The following are some of the publicly available repositories of SRS available.

Table 3: List of Web repositories containing publicly available Requirement documents

| Source | Description |
|---|---|
| https://kaggle.com | Contains datasets of Software requirements that are already labeled as Functional, Legal, Available, and so on. |
| https://worldwidescience.org | global science gateway comprised of national and international scientific databases and portals. Consists of more than 400 sample records of software requirements specification documents. |
| http://fmt.isti.cnr.it/nlreqdataset/ | A public dataset of Software Requirements specification documents given by the formal methods and tools group, Institute of Information Science and Technologies (ISTI), Italian National Research Council (CNR). |
| https://zenodo.org/record/1414117#.YV21BZpBw2w | The dataset here is PURE (Public REquirements dataset), a dataset of 79 publicly available natural language requirements documents collected from the Web [32] |

3.5 Summary of Related Work:

Table 4: Summary of findings from 2009-2021 presented by various authors

| Sno | Author/s | Year | Findings/Results/Interpretations |
|-----|---------------------------|------|---|
| 1. | Massey, A. K., et al [33] | 2009 | In this paper, a numerical assignment technique is proposed to prioritize legal documents. To implement, EHR systems are taken into consideration |
| 2. | Babar, M.I., et al. [34] | 2011 | An extension of Value based Intelligent Requirement Prioritization (VIRP) is proposed for Prioritization |
| 3. | Lau, K. K., et al [35] | 2011 | The authors have presented a very good heuristic approach for extracting keywords from Natural language requirements with the help of the POS tagging process. These extracted components can further be used to build COCOME systems. |
| 4. | Thakurta, R. [36] | 2012 | A quantitative framework is proposed to prioritize quality requirements. Stakeholders from organizations and developers are also included |
| 5. | Vlas, R. E., et al [37] | 2012 | The authors have proposed to develop a method in the future that will effectively classify software requirements of Open Source Software Development projects which will lead to a better understanding of requirements, their goals and objectives, and flow. They also have said that three specific contributions are required from NLP research to achieve this for OSSD. These three contributions are a grammar based design, an effective parsing scheme, and an efficient discovery and classification mechanism. |
| 6. | Voola, P., et al. [38] | 2013 | Discussion on different techniques of Requirements Prioritization by using Scales of measurement from Nominal to Ratio is proposed by the authors in this paper. |
| 7. | Tonella, P., et al [39] | 2013 | The consequence of having pairs of requirements is discussed in the paper. The authors have also used a |

| | | | |
|-----|------------------------------|------|--|
| | | | genetic algorithm to combine and process requirements with all information available in SRS regarding dependencies and priorities |
| 8. | Dabbagh, M., et al. [40] | 2014 | In this paper, an integrated prioritization approach (IPA) is introduced which can prioritize functional and non-functional requirements at the same time or concurrently, thus producing two prioritized separate lists of functional requirements and non-functional requirements separately. |
| 9. | Devulapalli, S., et al [41] | 2014 | A framework is suggested to prioritize requirements which take care of prioritization in successive releases of software by the authors |
| 10. | Shehzad, K., et al [42] | 2014 | Here, the theoretical aspect of Requirement Prioritization is discussed with respect to Requirement Engineering. A framework for a case study on Video Renting systems is proposed. The authors have considered minute factors and also the complex nature of requirements. |
| 11. | Khan, J.A., et al. [43] | 2015 | Through experiments, it is found that pairwise comparison (AHP) is more accurate, faster, and helpful than the numerical assignment. |
| 12. | Jawale, B., et al[44] | 2015 | In this paper, the authors say that it is not easy to deviate from customer expectations. They also say that it can be improved by Adaptive Fuzzy Hierarchical Cumulative Voting technique using fuzzy logic. |
| 13. | Gazi, Y., et al [45] | 2015 | In this paper, the authors have used multi-criteria decision-making methodologies to come up with a solution for Non-functional requirements. |
| 14. | Van Der Veen, E., et al [46] | 2015 | Pull requests automation has been discussed in this paper. A tool called PRioritizer is designed and implemented here. This tool gives another view of GitHub also. The authors have used Logistic Regression, Naïve Bayes, and Random Forest for a comparative analysis of the tool. They have also found out that Random Forest gives better results |
| 15. | McZara, J., et al [47] | 2015 | Software Prioritization problems for large scale projects are discussed here. A method called SNIPR is proposed which incorporates both NLP and satisfiability modulo theories (SMT) solvers for requirement prioritization process and also proved that it is efficient to combine technologies to achieve good results. |
| 16. | Babar, M. I., et al [48] | 2015 | In this research paper, an expert decision support system called the PHandler is developed to prioritize large scale requirements efficiently. It is an expert knowledge based system that combines three approaches such as VIRP, BPNN, and AHP. The results show that PHandler is suitable for the prioritization of requirements of large projects. |
| 17. | Kilimci, Z. H., et al [49] | 2015 | Unlike the traditional way of using Naïve Bayes for text classification, the authors have tried to use the multivariate Bernoulli model and also observed that the results are highly satisfying when combined with an appropriate smoothing method. |
| 18. | Nguyen, D. D., et al [50] | 2015 | In this paper, a method for text classification has been proposed which unlike the traditional methods can handle short, ambiguous, and written in non standard English |

| | | | |
|-----|---------------------------------|------|---|
| | | | language social messages. The authors also have to say that a good understanding of the characteristics aids in better classification |
| 19. | Pfaff, M., et al [51] | 2015 | The authors concentrate on IT benchmarking as the foundation for ontology creation. Further, they say that NLP and ML techniques can be used for this in the form of supplying data for ontology creation as the data is available in the form of semi structured and unstructured format. |
| 20. | Dabbagh, M., et al. [52] | 2016 | Comparing integrated prioritization approach (IPA) with Analytic hierarchy process (AHP)-based approach and IPA with Hybrid assessment method (HAM) -based approach |
| 21. | Setiani, N., et al [53] | 2016 | Content Management systems contain a lot of text which needs to be prioritized to process them for good results. In this paper, an approach is proposed using numerical assignment and clustering techniques. The prioritization technique is mostly based on the level of importance by using a matrix that is transformed by the normalization method. The lower the average value of the cluster, the higher is the importance level of requirement. |
| 22. | Yaseen, M., et al. [54] | 2017 | Requirement Prioritization is demonstrated with the help of a directed graph when one requirement is dependent on a previous one. |
| 23. | Gerogiannis, V. C., et al [55] | 2017 | An approach to prioritize requirements is discussed in the paper by considering uncertain ratings and rankings by multiple stakeholders by collective evaluations of all stakeholders. This is calculated and evaluated by using fuzzy linguistic 2-tuples and belongs to ordinal scale taxonomy. |
| 24. | Serral, E., et al [56] | 2017 | The authors have proposed a technique to prioritize contextual requirements which are nothing but Non-Functional Requirements with special reference to three Non-Function requirements such as Comfort, Efficiency, and Utility bill saving of a Smart Home. The technique was tested with the help of a structured questionnaire posted on the web. |
| 25. | Aliwy, A. H., et al [57] | 2017 | Five algorithms have been discussed here which are Decision Tree, Support Vector Machine, K-Nearest Neighbours, Naïve Bayes, and hidden Markov model. Some improvements such as modification, modification, and addition, and extraction and reduction are then made to each of these and they are analyzed. It is also found that feature reduction is the best for text classification. |
| 26. | Navarro-Almanza, R., et al [58] | 2017 | It is discussed in this paper that Deep Learning is proved to be more useful in Software Requirements classification. This also seems to reduce a lot of human effort. The authors further suggest that this technique can be used to process students home work in the future. Also, the results obtained are much lesser than those obtained through NLP. |
| 27. | Hudaib, A., et al. [7] | 2018 | Discussion and classification of various Software Requirements Prioritization techniques |
| 28. | Masadeh, R., et al. [59] | 2018 | Whale and Grey wolf optimization algorithms (WGW) have been proposed by the authors by combining whale |

| | | | |
|-----|---------------------------|------|---|
| | | | optimization algorithm and Grey wolf optimization algorithm by taking the best of both the algorithms to do prioritization |
| 29. | Bakly, K.E., et al [60] | 2018 | A fuzzy version of Enhancing Hybrid Requirements Prioritization technique which also takes care of unclear and uncertain data has been proposed which is a combination of QFD, CV, and AHP. |
| 30. | Ahuja H., et al [61] | 2018 | According to the authors, out of the nine techniques mentioned for prioritization process, VOP (value-oriented prioritization) is regarded as the best since it gives non-erroneous results and consists of easy-to-use method which also reduces the consumption of time. This paper also presents a new technique by using least-squares-based random genetic algorithm as compared to the earlier interactive genetic approach. |
| 31. | Felfernig, A., et al [62] | 2018 | The authors have focussed on utility based Requirement prioritization with special reference to open source projects. Here two scenarios are discussed where either one stakeholder has to complete the prioritization process or multiple users in a group need to do so. BUGZILLA is used to prioritization tasks. |
| 32. | Liu, H., et al [63] | 2018 | A very good method is proposed which not only classifies text but also can generate detailed explanations for decisions. To achieve this, summaries of textual matter and rating scores which are in numbers are taken into consideration. |
| 33. | Tufail, H., et al [64] | 2019 | In this paper, the authors discuss about 10 important Requirement Prioritization techniques with the help of a case study consisting of 12 requirements. It is also found that a combination of Cumulative Voting and Wieggers method were ranked high. |
| 34. | Zahoor T., et al [65] | 2019 | In this paper, a novel method to prioritize software requirements especially related to model driven engineering is discussed. UML profile to do Analytic Hierarchy Process is described. |
| 35. | Jahan, M. S., et al [66] | 2019 | The authors have proposed a new method called “MAHP” where the requirements are categorized into four groups by using MOSCOW technique. Later AHP is applied to prioritize them. MAHP is tested on a Library management case study. |
| 36. | Din, J., et al [67] | 2019 | Human inconsistency in judging requirements becomes a problem as the volume of requirements increases. A hybrid prioritization technique tool to prioritize requirements has been proposed which uses AHP and other cost value methods. Hence AHP is used to calculate the inconsistency. |
| 37. | Yaseen, M., et al [68] | 2019 | Here requirements that are large in size have been considered. An iterative model of Requirement Prioritization has been suggested using which delay in executing requirements can be stopped. The authors also suggest that requirements that are independent and do not require the implementation of other requirements need to be implemented first to save time and effort, after which dependent requirements can be implemented. |

| | | | |
|-----|---------------------------------|------|--|
| 38. | Barbosa, P. A., et al [69] | 2019 | A methodology that uses Verbal Decision Analysis (VDA) has been introduced in the paper. This proposed method can support decision making in selecting and prioritizing requirements. This has been achieved by using DMs, metaheuristics, and ORCLASS method. |
| 39. | Haider, W., et al [70] | 2019 | A framework for requirement prioritization and ranking and tracing for global software development (GSD) using the AI technique is proposed and discussed in this paper. Team Foundation Server (TFS) for communication and Case based Ranking (CBR) to find the ranking. J48 algorithm is also used for classification. |
| 40. | Khan, A. I., et al [71] | 2019 | In this paper, the authors have dealt with prioritizing requirements for M-learning applications. As the available techniques of requirements prioritization do not suit M-learning applications, the authors have proposed four step process requirements prioritization approach through a case study on Computer Science course learning applications. The authors have found that it has yielded positive results about the approach being feasible. |
| 41. | Luo, L., et al [72] | 2019 | The authors have developed a machine learning approach to prioritize the admission process in hospitals thus helping doctors to take a suitable decision regarding which patient requires admission. Logistic Regression, Random Forest, Gradient Boosting Decision Tree, Extreme gradient boosting (XGBoost), and an ensemble model have been used for the purpose and XGBoost is proved to have given the best results |
| 42. | Kallis, R., et al [73] | 2019 | In this work, the authors have presented an app namely Ticket Tagger which can automatically categorize GitHub projects as a bug report, request, or question. This is done with the help of a machine learning model. |
| 43. | Carchiolo, V., et al [74] | 2019 | The authors have developed an application to classify Medical prescriptions based on NLP and Image Processing techniques. The classification helps the operator to automatize scrutinizing of prescriptions and classify them. |
| 44. | Roy, M., et al [75] | 2020 | An approach to prioritize Non Functional Requirements has been proposed which otherwise poses lots of inconsistencies. The proposed algorithm works based on the weights assigned by the analyst. |
| 45. | Checa Cabrera M. A., et al [76] | 2020 | This paper uses static analysis of Neutrosophic Cognitive Maps to prioritize Non-Functional Requirements for a mobile application called panic button in Equador. |
| 46. | Yaseen, M., et al [77] | 2020 | In this paper, the authors have used Directed acyclic graphs (DAG) to assign priorities to requirements. Furthermore, prioritization is done by using certain protocols. The priority is assigned based on the dependency of other requirements on any given requirement. This method is tested on 27 functional requirements collected from a mobile sales shop. |
| 47. | Cheng, F.Y., et al [78] | 2020 | In this paper, the authors have used Machine Learning algorithms to prioritize Covid 19 patients who have to be transferred to ICU based on certain parameters or variables. To achieve this, they have developed a novel |

| | | | |
|-----|----------------------------|------|---|
| | | | supervised machine learning classifier using EMR data and applied the Random Forest approach. |
| 48. | Segarra-Faggioni, V., [79] | 2020 | The authors have proposed a suitable method to classify annotated reports of students by using SVM and RFC. Also, it is concluded that Machine learning algorithms and NLP are very helpful for the said task |
| 49. | Prasad, S. [80] | 2020 | The author has explored ways to classify complaints received by using Information technology techniques especially Natural Language Processing techniques. He also researches on possibilities of Linguistic style features for improving classification. |
| 50. | Arteaga, C., et al [81] | 2020 | In this paper, the authors propose an approach to identify various factors associated with respect to severity of injury during vehicle accidents. This is done with Text mining aided by Machine Learning. |
| 51. | Wijesinghe, D., et al [82] | 2020 | The authors have tried to find out ways to order and prioritize journalistic notes by taking up a theme. To do this, Python Programming language is used and also in order to evaluate, similarity and ranking algorithms of NLP are employed. |
| 52. | Das, R. K., et al [83] | 2020 | Afinn and Bing are the two lexicons used in this paper to carry out the task. Scores are assigned and calculated for all the grievance and then categorised as high, medium and low priority grievances. Categorising grievances ensures good and timely service by Government authorities. |
| 53. | Nasr, M., et al [84] | 2020 | Topic modelling and Topic classification are discussed here. SVM is used Topic classification. LSA, LDA, PLSA algorithms are applied for Topic modelling |
| 54. | Ahmed, H. A., et al [85] | 2021 | CaPBug is a framework that can categorize and prioritize bug reports. The tool has been tested on bug reports of Eclipse and Mozilla from the Bugzilla dataset. NLP techniques for classification and supervised machine learning and machine learning classifiers were used for the purpose. |
| 55. | Halder, M., et al [86] | 2021 | The authors have devised a method for Email notification system by identifying and predicting important 'n' mails for notification at regular intervals of time. This has been made possible with the help of Natural Language Processing algorithms. |
| 56. | Prasad G. N. R [87] | 2021 | In this paper, the author has used NLP to identify the various Taxonomy levels of Blooms Taxonomy in the given question paper based on the keywords of the questions. |
| 57. | Shafiq, S., et al [88] | 2021 | The authors have proposed an approach using NLP for Issues prioritization which will be helpful to Project managers to plan out the execution of newly added or modified issues. |

4. DISCUSSION AND FUTURE WORK :

The review done shows that the existing Software Requirements Prioritization methods are not automated and they are done manually by using any of the methods that are mentioned. At the same time, it has been observed that Text Prioritization can be done with the help of Machine Learning Algorithms [89]. This has already been done in the Medical field to prioritize the treatment of patients

based on their health records [90], [91]. Moreover, there are no specific methods for Requirements Prioritization during Software Development based on the type of application being developed [92]. Hence, there is a need to find out a way to develop an automated way of Prioritizing requirements in general and for every type of application being developed.

4.1 Proposed outline of Prioritizing requirements using NLP:

The following steps can be proposed in Prioritizing requirements:

- a) All Software Requirement Specification documents will be stored in a repository.
- b) All the specification documents will undergo pre-processing. This is done as a part of Data cleaning
- c) The documents will be classified into different categories by using appropriate Machine Learning algorithms [93]
- d) Weights can be applied to each document based on the keywords used for classification
- e) All the classified documents can be rearranged based on the ML algorithms by finding the probability of each document [94]
- f) Priority can be set to thus arranged documents

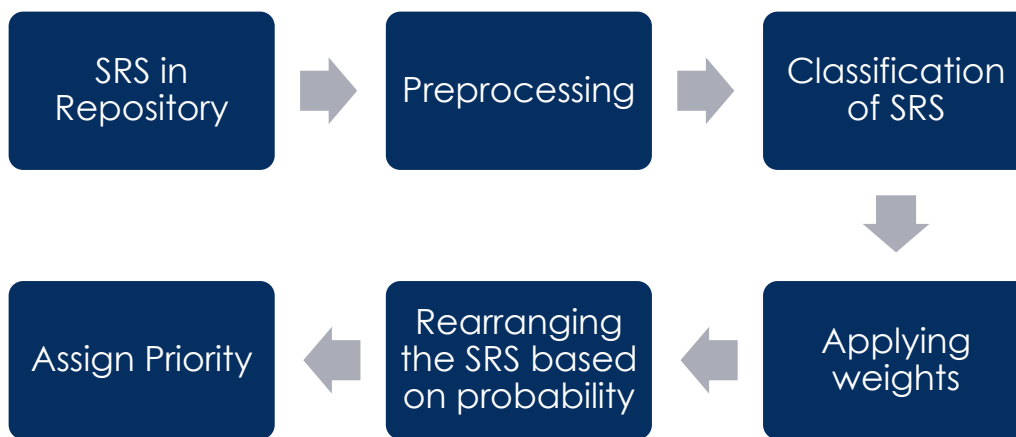


Fig. 3: Block diagram of steps proposed in Requirements prioritization method

5. RESEARCH GAP :

The Literature survey conducted with the identified keywords has shown various Requirement Prioritization methods. All these methods and techniques are suitable to prioritize requirements manually. There is no mention of automated prioritization of requirements. At the same time, instances of text prioritization using ML algorithms are present [95]. The following Research Gaps are identified-

Research Gap 1: Which of the available SRP methods can be automated?

Out of so many Prioritization methods identified, which ones are suitable to be chosen for an automation process. It has to be evaluated based on efficiency and diligence.

Research Gap 2: Many categories and types of algorithms are available in NLP and Machine Learning [96]. Out of these which one can be applied for SRP?

Different algorithms and methods are available in ML and NLP for classification and assigning priority. Which one/s among them will serve the purpose and needs to be worked on.

Research Gap 3: Are there different Prioritization techniques available for different types of applications to be built?

Though there are no different ways of building a software application, it has to be seen if requirement prioritization methods can be different for various types of software applications.

Research Gap 4: Are there any other repositories for SRS other than the ones mentioned in this literature survey?

There are very few public repositories that are available for SRS. Are there any means of finding out a few more for the purpose of research?

6. RESEARCH AGENDAS :

- i. What are the different techniques/algorithms available for prioritizing software requirements?
- ii. What are the characteristics and drawbacks of the existing software requirements prioritization techniques?
- iii. What are the technologies under study to automate the process of software requirements prioritization process?
- iv. What is the extent of research carried out in NLP and ML for text Prioritization?
- v. What are the techniques available/suitable for Software Requirements Prioritization using Natural Language Processing and Machine Learning?
- vi. What are the Open Source NLP tools available?
- vii. Which are the available repositories for Software Requirement Specification Documents?

7. ANALYSIS OF RESEARCH AGENDAS :

Different techniques are available for prioritizing requirements. Each one has its own merits and demerits. The purpose of using each of the techniques also varies. All the techniques available give a basic guideline or framework that can be applied to prioritize requirements. Hence a question arises as to how these techniques can be automated [97]. AI and ML algorithms can fit into some of the techniques and help in achieving prioritization tasks. A fair amount of research has already gone into text prioritization in the area of NLP and ML. This research has to be kept as a foundation or roadmap to further enhance it and apply it for Software Requirements. There are lots of open source tools available for working with Software Requirements. It has to be seen which among these will serve the purpose of Prioritization. Also, a suitable repository of Software Requirements Specification Documents has to be identified so that the effectiveness of the technique can be seen.

8. FINAL RESEARCH PROPOSAL IN CHOSEN TOPIC :

Automation of Requirement prioritization is something which the Software Industry would be looking for due to the ever increasing requirements and also the difficulties involved in prioritizing them [98]. There are very few pieces of evidence and research done on the automation process of prioritizing requirements. Lots of medical documents have been prioritized and subsequently used in the research thereof [99]. Similarly, the Software Requirements Specification document can be used as a source and the classified requirements can be prioritized by using NLP techniques backed by ML algorithms. On the first trial, this can be done for any type of applications. Hence, the closest topic could be- **Implementation of Automated Software Requirements Prioritization by using NLP techniques**

9. SLOC ANALYSIS OF RESEARCH PROPOSAL :

SLOC analysis is done to analyze and assess Strengths, Limitations, Opportunities, and Challenges [100] [101]. The following is the result of the SLOC analysis conducted on the Research proposal.

Table 5: SLOC analysis

| Strengths | Limitations |
|--|---|
| <ul style="list-style-type: none"> • Availability of predefined techniques for Requirement Prioritization • Appropriate availability of technology in the form of NLP and ML • Availability of various tools to analyze and prioritize • Upcoming and most sought-after Research Problem in the domain of Software Engineering | <ul style="list-style-type: none"> • Handling unstructured data is a big challenge • No standard format is followed by all while gathering Requirements • The reliability and authenticity of data need to be ascertained. |
| Opportunities | Challenges |

| | |
|--|---|
| <ul style="list-style-type: none">• Requirement Prioritization is an issue that is most sought after by the Software companies• This issue also opens up new avenues for Research | <ul style="list-style-type: none">• Legal issues and corporate laws govern the data or Requirement Specification documents• Requirement Specification documents do not follow a uniform structure• Very few repositories of Requirement Specification documents are available |
|--|---|

10. CONCLUSION :

Tackling and handling issues in Software Development has always been challenging. Especially, the timely delivery of developed software to the stakeholders with all requirements fulfilled is commendable. Fulfilling all requirements in an ordered way is the heart and soul of Software Development. Therefore, deciding the order in which the requirements need to be processed takes an upper hand in Software Engineering Research. There are many Requirement Prioritization techniques available. But all these techniques need to be carried out by the development team manually which is a cumbersome process and also bias prone. Hence, there is a need to automate the process of Requirement Prioritization. Literature review done in this regard shows that Natural Language Processing ably supported by Machine Learning if required can be used to achieve the task. Similar research has been studied in this regard. Appropriate Machine Learning Algorithms have to be analyzed for this purpose. A comparative study needs to be done in this regard to find out the best algorithm. Or if the need arises a new algorithm or method can be devised.

REFERENCES :

- [1] Chomal, V. S., & Saini, J. R. (2015). On Prioritizing and Freezing of Software Requirements. *International Journal of Innovative Science, Engineering & Technology*, 2(11), 807-817.
[Google Scholar](#)
- [2] Xiao, B., Su, K., & Su, X. (2020). A Requirement Engineering Approach to User-Centered Design. *Journal of Physics: Conference Series*, 1453(1), 1-6.
[Google Scholar](#) [CrossRef/DOI](#)
- [3] Singh, S., & Saikia, L. P. (2016). Formal Methods and its Importance in Minimizing Ambiguity in the Requirement Engineering Document Phase of SDLC. *Journal of Applied Information Science*, 4(2), 26-32.
[Google Scholar](#)
- [4] Yaseen, M., Mustapha, A., & Ibrahim, N. (2019). Prioritization of software functional requirements: spanning tree based approach. *International Journal of Advanced Computer Science and Applications*, 10(7), 489-497.
[Google Scholar](#)
- [5] Goel, L. B., & Thakur, S. (2013) Requirement Prioritization: A Study and Proposed Framework. *International Journal of Emerging Technology and Advanced Engineering*, 3(6), 2250-2459.
[Google Scholar](#)
- [6] Varun, G., Shivam, L., Deniz, Ç., & Hye-jin, K. (2017). Non-functional Requirement Prioritization Approach. *International Journal of Software Engineering and Its Applications*, 11(1), 61-66.
[Google Scholar](#)
- [7] Hudaib, A., Masadeh, R., Qasem, M. H., & Alzaqebah, A. (2018). Requirements prioritization techniques comparison. *Modern Applied Science*, 12(2), 62-80.
[Google Scholar](#) [CrossRef/DOI](#)
- [8] Asghar, A. R., Bhatti, S. N., Tabassum, A., & Shah, S. A. A. S. (2017). The impact of analytical assessment of requirements prioritization models: an empirical study. *International Journal of Advanced Computer Science and Applications*, 8(2), 303-313.
[Google Scholar](#)

- [9] AbdElazim, K., Moawad, R., & Elfakharany, E. (2020). A framework for requirements prioritization process in agile software development. *Journal of Physics: Conference Series*, 1454(1), 1-11.
[Google Scholar](#) [CrossRef/DOI](#)
- [10] Olaronke, I., Rhoda, I., & Ishaya, G. (2018). An Appraisal of Software Requirement Prioritization Techniques. *Asian Journal of Research in Computer Science*, 1(1), 1-16.
[Google Scholar](#) [CrossRef/DOI](#)
- [11] Perini, A., Susi, A., & Avesani, P. (2013). A Machine Learning Approach to Software Requirements Prioritization. *IEEE Transactions on Software Engineering*, 39(4), 445-461.
[Google Scholar](#) [CrossRe/DOI](#)
- [12] Ejaz, K., & Amjad, A. (2018). Model and Technique over Software Requirement Prioritization. *Pakistan Journal of Engineering, Technology & Science*, 6(2), 156-167.
[Google Scholar](#) [CrossRef/DOI](#)
- [13] Choudhari, S., Choudhary, N., Kaware, S., & Shaikh, A. (2020). Email Prioritization Using Machine Learning. SSRN Electronic Journal. Published.
[Google Scholar](#) [CrossRef/DOI](#)
- [14] Singh, M., Murthy, A., & Singh, S. (2015). Prioritization of free-text clinical documents: a novel use of a bayesian classifier. *JMIR medical informatics*, 3(2), e17.
[Google Scholar](#) [CrossRef/DOI](#)
- [15] Amora, P. R. P., et al (2018). An Analysis of Machine Learning Techniques to Prioritize Customer Service Through Social Networks. *Journal of Information and Data Management*, 9(2), 135-146.
[Google Scholar](#)
- [16] Muqem, M., & Beg, M. R. (2015). A Fuzzy Based Approach for Early Requirement Prioritization. *International Journal of Computers & Technology*, 15(2), 6480-6490.
[Google Scholar](#)
- [17] Sharma, N., & Yalla, P. (2018). Developing research questions in natural language processing and software engineering. *JOIV: International Journal on Informatics Visualization*, 2(4), 268-270.
[Google Scholar](#) [CrossRef/DOI](#)
- [18] Vibha, G., & Anuja, S. (2011). Identifying an Appropriate Requirements Prioritization Methodology Using Fuzzy Decision-Making. *International Conference on Information Processing*, 157(1), 258-268
[Google Scholar](#)
- [19] Salado, A., & Nilchiani, R. (2015). Adaptive Requirements Prioritization (ARP): Improving Decisions between Conflicting Requirements. *Systems Engineering*, 18(5), 472-490.
[Google Scholar](#) [CrossRef](#)
- [20] Yalla, P., & Sharma, N. (2015). Integrating natural language processing and software engineering. *International Journal of Software Engineering and Its Applications*, 9(11), 127-136.
[Google Scholar](#) [CrossRef](#)
- [21] Nazir, F., Butt, W. H., Anwar, M. W., et al (2017). The applications of natural language processing for software requirement engineering-a systematic literature review. *2017 International conference on information science and applications*, Online ISBN 978-981-10-4154-9, 485-493. Springer.
[Google Scholar](#) [CrossRef](#)
- [22] Abdelnabi, E. A., Maatuk, A. M., Abdelaziz, T. M., & Elakeili, S. M. (2020). Generating UML Class Diagram using NLP Techniques and Heuristic Rules. *2020 20th International Conference on Sciences and Techniques of Automatic Control and, Computer Engineering*. Electronic ISBN:978-1-7281-8815-7. 277-282. IEEE.
[Google Scholar](#) [CrossRef/DOI](#)
- [23] Bhardwaj, A., Khanna, P., & Kumar, S. (2020). Generative Model for NLP Applications based on Component Extraction. *Procedia Computer Science*, 167, 918-931.
[Google Scholar](#) [CrossRef/DOI](#)

- [24] Tatwadarshi P. N. — April 23, 2021. Role of Machine Learning in Natural Language Processing. <https://www.analyticsvidhya.com/blog/2021/04/role-of-machine-learning-in-natural-language-processing/>. Retrieved on 15th September 2021.
- [25] Natural Language Toolkit . Retrieved from <https://www.nltk.org/> on 17th September 2021.
- [26] CoreNLP . Retrieved from <https://stanfordnlp.github.io/CoreNLP/> on 17th September 2021.
- [27] OpenNLP. Retrieved from <https://opennlp.apache.org/> on 17th September 2021.
- [28] Mallet. Retrieved from <http://mallet.cs.umass.edu/> on 17th September 2021.
- [29] GATE. Retrieved from <https://gate.ac.uk/> on 17th September 2021.
- [30] spaCy. Retrieved from <https://spacy.io/> on 17th September 2021.
- [31] Genism 4.1.2. Retrieved from <https://pypi.org/project/gensim/> on 17th September 2021.
- [32] Ferrari, A., Spagnolo, G. O., & Gnesi, S. (2017, September). PURE: A dataset of public requirements documents. *2017 IEEE 25th International Requirements Engineering Conference (RE)*, Electronic ISBN:978-1-5386-3191-1, 502-505. IEEE.
[Google Scholar↗](#) [CrossRef/DOI↗](#)
- [33] Massey, A. K., Otto, P. N., & Antón, A. I. (2009). Prioritizing legal requirements. *2009 Second International Workshop on Requirements Engineering and Law*, Electronic ISBN:978-0-7695-4102-0, 27-32. IEEE.
[Google Scholar↗](#) [CrossRef/DOI↗](#)
- [34] Babar, M. I., Ramzan, M., & Ghayyur, S. A. (2011, July). Challenges and future trends in software requirements prioritization. *International conference on computer networks and information technology*, Electronic ISBN:978-1-61284-941-6, 319-324. IEEE.
[Google Scholar↗](#) [CrossRef/DOI↗](#)
- [35] Lau, K. K., Nordin, A., & Ng, K. Y. (2011). Extracting elements of component-based systems from natural language requirements. *37th EUROMICRO Conference on Software Engineering and Advanced Applications*, Print ISBN:978-1-4577-1027-8, 39-46. IEEE.
[Google Scholar↗](#) [CrossRef/DOI↗](#)
- [36] Thakurta, R. (2013). A framework for prioritization of quality requirements for inclusion in a software project. *Software Quality Journal*, 21(4), 573-597.
[Google Scholar↗](#) [CrossRef/DOI↗](#)
- [37] Vlas, R. E., & Robinson, W. N. (2012). Two rule-based natural language strategies for requirements discovery and classification in open source software development projects, *Journal of management information systems*, 28(4), 11-38.
[Google Scholar↗](#) [CrossRef/DOI↗](#)
- [38] Voola, P., & Babu, A. V. (2013). Comparison of requirements prioritization techniques employing different scales of measurement. *ACM SIGSOFT Software Engineering Notes*, 38(4), 1-10.
[Google Scholar↗](#) [CrossRef/DOI↗](#)
- [39] Tonella, P., Susi, A., & Palma, F. (2013). Interactive requirements prioritization using a genetic algorithm. *Information and software technology*, 55(1), 173-187.
[Google Scholar↗](#) [CrossRef/DOI↗](#)
- [40] Dabbagh, M., & Lee, S. P. (2014). An approach for integrating the prioritization of functional and nonfunctional requirements. *The Scientific World Journal*, 2014. 1-13.
[Google Scholar↗](#) [CrossRef/DOI↗](#)
- [41] Devulapalli, S., & Khare, A. (2014). A Framework for Requirement Prioritization for Software Products. *IUJ Journal of Management*, 2(1), 35-41.
[Google Scholar↗](#)

- [42] Shehzad, K., Awan, M. D., Rizvi, S. S., & Khiyal, S. H. (2014). A hybrid technique based on standard SRS modules for software requirement prioritization. *Proceedings of Informing Science & IT Education Conference (InSITE) 2014*, ISSN: 1535-0703, 279-294.
[Google Scholar](#) [CrossRef/DOI](#)
- [43] Khan, J. A., Rehman, I. U., Khan, Y. H., Khan, I. J., & Rashid, S. (2015). Comparison of Requirement Prioritization Techniques to Find Best Prioritization Technique. *International Journal of Modern Education & Computer Science*, 7(11), 53-59.
[Google Scholar](#) [CrossRef/DOI](#)
- [44] Jawale, B., & Bhole, A. T. (2015). Adaptive fuzzy hierarchical cumulative voting: a novel approach toward requirement prioritization. *International Journal of Research in Engineering and Technology*, 4(05), 365-370.
[Google Scholar](#)
- [45] Gazi, Y., & Sadiq, M. (2015). Prioritization of NFRs using Multi-Criteria Decision-Making Methods. *International Journal of Computer Applications*, 123(3), 6-10.
[Google Scholar](#)
- [46] Van Der Veen, E., Gousios, G., & Zaidman, A. (2015, May). Automatically prioritizing pull requests. *2015 IEEE/ACM 12th Working Conference on Mining Software Repositories*, Electronic ISBN:978-0-7695-5594-2, 357-361. IEEE.
[Google Scholar](#) [CrossRef/DOI](#)
- [47] McZara, J., Sarkani, S., Holzer, T., & Eveleigh, T. (2015). Software requirements prioritization and selection using linguistic tools and constraint solvers—a controlled experiment. *Empirical Software Engineering*, 20(6), 1721-1761.
[Google Scholar](#) [CrossRef/DOI](#)
- [48] Babar, M. I., Ghazali, M., Jawawi, D. N., Shamsuddin, S. M., & Ibrahim, N. (2015). PHandler: an expert system for a scalable software requirements prioritization process. *Knowledge-Based Systems*, 84(1), 179-202.
[Google Scholar](#) [CrossRef/DOI](#)
- [49] Kilimci, Z. H., & Ganiz, M. C. (2015, September). Evaluation of classification models for language processing. *2015 International Symposium on Innovations in Intelligent Systems and Applications (INISTA)*, Electronic ISBN:978-1-4673-7751-5, 1-8. IEEE.
[Google Scholar](#) [CrossRef/DOI](#)
- [50] Nguyen, D. D., Dao, M. S., & Nguyen, T. V. T. (2015). Natural language processing for social event classification. *Knowledge and Systems Engineering*, Online ISBN 978-3-319-11680-8, 79-91. Springer, Cham.
[Google Scholar](#) [CrossRef/DOI](#)
- [51] Pfaff, M., & Krcmar, H. (2015). Natural Language Processing Techniques for Document Classification in IT Benchmarking. *Proceedings of the 17th International Conference on Enterprise Information Systems-Volume 1*, ISBN: 978-989-758-096-3, 360-366.
[Google Scholar](#) [CrossRef/DOI](#)
- [52] Dabbagh, M., Lee, S. P., & Parizi, R. M. (2016). Functional and non-functional requirements prioritization: empirical evaluation of IPA, AHP-based, and HAM-based approaches. *Soft computing*, 20(11), 4497-4520.
[Google Scholar](#) [CrossRef/DOI](#)
- [53] Setiani, N., & Dirgahayu, T. (2016). Clustering technique for information requirement prioritization in specific CMSs. *2016 International Conference on Data and Software Engineering (ICoDSE)*. Electronic ISBN:978-1-5090-5671-2, 1-6. IEEE.
[Google Scholar](#) [CrossRef/DOI](#)
- [54] Yaseen, M., Mustapha, A., Rahman, A. U., Khan, S., & Kamal, W. (2020). Importance of requirements prioritization in parallel developing software projects. *International Journal of Scientific Research in Computer Science and Engineering*, 9(2), 171-179.

[Google Scholar](#)↗

- [55] Gerogiannis, V. C., & Tzikas, G. (2017). Using Fuzzy Linguistic 2-Tuples to Collectively Prioritize Software Requirements based on Stakeholders' Evaluations. *Proceedings of the 21st Pan-Hellenic Conference on Informatics 2017*. 1-6.
[Google Scholar](#)↗ [CrossRef/DOI](#)↗
- [56] Serral, E., Sernani, P., Dragoni, A. F., & Dalpiaz, F. (2017, April). Contextual requirements prioritization and its application to smart homes. *European conference on ambient intelligence*, Online ISBN 978-3-319-56997-0, 94-109. Springer, Cham.
[Google Scholar](#)↗ [CrossRef/DOI](#)↗
- [57] Aliwy, A. H., & Ameer, E. A. (2017). Comparative study of five text classification algorithms with their improvements. *International Journal of Applied Engineering Research*, 12(14), 4309-4319.
[Google Scholar](#)↗
- [58] Navarro-Almanza, R., Juarez-Ramirez, R., et al. (2017). Towards supporting software engineering using deep learning: A case of software requirements classification. *5th International Conference CONISOFT*, Electronic ISBN:978-1-5386-3956-6, 116-120. IEEE.
[Google Scholar](#)↗ [CrossRef/DOI](#)↗
- [59] Masadeh, R., Hudaib, A., & Alzaqebah, A. (2018). WGW: A Hybrid Approach Based on Whale and Grey Wolf Optimization Algorithms for Requirements Prioritization. *Advances in Systems Science and Applications*, 18(2), 63-83.
[Google Scholar](#)↗ [CrossRef/DOI](#)↗
- [60] Bakly, A. E., & Darwish, N. R. (2018). A Proposed Hybrid Prioritization Technique for Software Requirements based on Fuzzy Logic. *CiiT International Journal of Fuzzy Systems*, 10(2), 45-52.
[Google Scholar](#)↗
- [61] Ahuja, H., & Batra, U. (2018). Performance enhancement in requirement prioritization by using least-squares-based random genetic algorithm. *Innovations in Computational Intelligence*, Online ISBN 978-981-10-4555-4, 251-263, Springer, Singapore.
[Google Scholar](#)↗ [CrossRef/DOI](#)↗
- [62] Felfernig, A., Stettinger, M., Atas, M., Samer, R., Nerlich, J., Scholz, S., et al (2018). Towards utility-based prioritization of requirements in open source environments. *26th International Requirements Engineering Conference (RE)*, Electronic ISBN:978-1-5386-7418-5. 406-411. IEEE.
[Google Scholar](#)↗ [CrossRef/DOI](#)↗
- [63] Liu, H., Yin, Q., & Wang, W. Y. (2018). Towards explainable NLP: A generative explanation framework for text classification. arXiv preprint arXiv:1811.00196.
[Google Scholar](#)↗
- [64] Tufail, H., Qasim, I., Masood, M. F., Tanvir, S., & Butt, W. H. (2019, March). Towards the selection of optimum requirements prioritization technique: a comparative analysis. *5th International Conference on Information Management (ICIM)*, Electronic ISBN:978-1-7281-3430-7, 227-231. IEEE.
[Google Scholar](#)↗ [CrossRef/DOI](#)↗
- [65] Zahoor, T., Azam, F., Anwar, M. W., Maqbool, B., & Javaid, H. A. (2019). A UML Profile for Software Requirements Prioritization. *2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, Electronic ISBN:978-1-7281-2530-5, 885-891. IEEE.
[Google Scholar](#)↗ [CrossRef/DOI](#)↗
- [66] Jahan, M. S., Azam, F., Anwar, M. W., Amjad, A., & Ayub, K. (2019, October). A Novel Approach for Software Requirement Prioritization based upon non functional requirements. *7th International Conference in Software Engineering Research and Innovation (CONISOFT)*, Electronic ISBN:978-1-7281-2524-4, 1-7. IEEE.
[Google Scholar](#)↗ [CrossRef/DOI](#)↗

- [67] Din, J., Michael, I., & Jasser, M. B. (2019). Software Requirements Prioritization Tool using a Hybrid Technique. *International Journal of Engineering and Advanced Technology (IJEAT)*, 9(1), 1631-1635
[Google Scholar](#) [CrossRef/DOI](#)
- [68] Yaseen, M., Ibrahim, N., & Mustapha, A. (2019). Requirements prioritization and using iteration model for successful implementation of requirements. *Int. J. Adv. Comput. Sci. Appl*, 10(1), 121-127.
[Google Scholar](#)
- [69] Barbosa, P. A., Pinheiro, P. R., & Silveira, F. R. (2019). Selection and Prioritization of Software Requirements Applying Verbal Decision Analysis. *Complexity*, 2019, 1-20.
[Google Scholar](#) [CrossRef/DOI](#)
- [70] Haider, W., Hafeez, Y., Ali, S., Jawad, M., Ahmad, F. B., Rafi, M. N. (2019). Improving Requirement Prioritization and Traceability using Artificial Intelligence Technique for Global Software Development, *22nd International Multitopic Conference (INMIC)*. Electronic ISBN:978-1-7281-4001-8, 1-8. IEEE.
[Google Scholar](#) [CrossRef/DOI](#)
- [71] Khan, A. I., Khanjari, Z. A., & Sarrab, M. (2019). Prioritising mobile learning application requirements. *International Journal of Business Information Systems*, 32(1), 91-108.
[Google Scholar](#) [CrossRef/DOI](#)
- [72] Luo, L., Li, J., Liu, C., & Shen, W. (2019). Using machine-learning methods to support health-care professionals in making admission decisions. *The International journal of health planning and management*, 34(2), e1236-e1246.
[Google Scholar](#) [CrossRef/DOI](#)
- [73] Kallis, R., Di Sorbo, A., Canfora, G., & Panichella, S. (2019). Ticket tagger: Machine learning driven issue classification. *IEEE International Conference on Software Maintenance and Evolution (ICSME)*, Electronic ISBN:978-1-7281-3094-1, 406-409, IEEE.
[Google Scholar](#) [CrossRef/DOI](#)
- [74] Carchiolo, V., Longheu, A., Reitano, G., & Zagarella, L. (2019). Medical prescription classification: a NLP-based approach, *Federated Conference on Computer Science and Information Systems (FedCSIS)*, Electronic ISBN:978-83-952357-8-8, 605-609. IEEE.
[Google Scholar](#) [CrossRef/DOI](#)
- [75] Roy, M., Deb, N., Cortesi, A., Chaki, R., & Chaki, N. (2021). NFR-aware prioritization of software requirements. *Systems Engineering*, 24(3), 158-176.
[Google Scholar](#) [CrossRef/DOI](#)
- [76] Checa Cabrera, M. A., et al (2020). Prioritization of non-functional requirements in a mobile application for panic button system using neutrosophic decision maps. *Neutrosophic Sets and Systems*, 34(1), 153-158.
[Google Scholar](#)
- [77] Yaseen, M., Mustapha, A., Qureshi, S., Khan, A., & Rahman, A. U. (2020). A graph based approach to prioritization of software functional requirements. *International Journal of Computer Science and Software Engineering*, 9(4), 20-29.
[Google Scholar](#)
- [78] Cheng, F. Y., Joshi, H., Tandon, P., Freeman, R., Reich, D. L., Mazumdar, M., ... & Kia, A. (2020). Using machine learning to predict ICU transfer in hospitalized COVID-19 patients. *Journal of clinical medicine*, 9(6), 1668, 1-12.
[Google Scholar](#) [CrossRef/DOI](#)
- [79] Segarra-Faggioni, V., & Ratte, S. (2020, October). Computer-based classification of student's report. *12th International Conference on Education Technology and Computers*, ISBN 9781450388276, 33-36.
[Google Scholar](#) [CrossRef/DOI](#)

- [80] Prasad, S. (2020). Use of Natural Language Processing to Improve Complaint Classification in Customer Complaint Management System. *Journal of Critical Reviews*, 7(14), 2642-2652.
[Google Scholar](#) [CrossRef/DOI](#)
- [81] Arteaga, C., Paz, A., & Park, J. (2020). Injury severity on traffic crashes: A text mining with an interpretable machine-learning approach. *Safety Science*, 132, 104988, 1-12.
[Google Scholar](#) [CrossRef/DOI](#)
- [82] Wijesinghe, D., & Vidanage, K. (2020, March). Review On Approaches for Theme Extraction and Sentence Ordering for Prioritization of Journalistic Notes. *International Conference on Image Processing and Robotics (ICIP)*, Electronic ISBN:978-1-7281-6541-7, 1-6. IEEE.
[Google Scholar](#) [CrossRef/DOI](#)
- [83] Das, R. K., Panda, M., & Dash, S. S. (2020). Prioritizing Public Grievance Redressal Using Text Mining and Sentimental Analysis. *Advanced Computing and Intelligent Engineering: Proceedings of ICACIE 2018*, 1(1082), 273-284. Springer.
[Google Scholar](#)
- [84] Nasr, M., Karam, A., Atef, M., Boles, K., Samir, K., & Raouf, M. (2020). Natural Language Processing: Text Categorization and Classifications. *International Journal of Advanced Networking and Applications*, 12(2), 4542-4548.
[Google Scholar](#)
- [85] Ahmed, H. A., Bawany, N. Z., & Shamsi, J. A. (2021). CaPBug-A Framework for Automatic Bug Categorization and Prioritization Using NLP and Machine Learning Algorithms. Electronic ISSN: 2169-3536, *IEEE Access*, 9(1), 50496-50512.
[Google Scholar](#) [CrossRef/DOI](#)
- [86] Halder, M., Maheshwari, T., & Suresh, S. R. M. (2021). A Novel Approach to Control Emails Notification using NLP. *Procedia Computer Science*, 189, 224-231.
[Google Scholar](#) [CrossRef/DOI](#)
- [87] Prasad, G. N. R. (2021). Identification of Bloom's Taxonomy level for the given Question paper using NLP Tokenization technique. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, 12(13), 1872-1875.
[Google Scholar](#)
- [88] Shafiq, S., Mashkoo, A., Mayr-Dorn, C., & Egyed, A. (2021, September). NLP4IP: Natural Language Processing-based Recommendation Approach for Issues Prioritization. *47th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, Electronic ISBN:978-1-6654-2705-0, 99-108. IEEE.
[Google Scholar](#) [CrossRef/DOI](#)
- [89] Alenezi, M., & Banitaan, S. (2013, December). Bug reports prioritization: Which features and classifier to use?. *12th International Conference on Machine Learning and Applications*, Electronic ISBN:978-0-7695-5144-9, 112-116). IEEE.
[Google Scholar](#) [CrossRef/DOI](#)
- [90] King, A. J., Cooper, G. F., Clermont, G., Hochheiser, H., Hauskrecht, M., Sittig, D. F., & Visweswaran, S. (2020). Leveraging eye tracking to prioritize relevant medical record data: comparative machine learning study. *Journal of medical Internet research*, 22(4), e15876.
[Google Scholar](#) [CrossRef/DOI](#)
- [91] Spasic, I., & Nenadic, G. (2020). Clinical text data in machine learning: systematic review. *JMIR medical informatics*, 8(3), e17984.
[Google Scholar](#) [CrossRef/DOI](#)
- [92] Awais, M. A. (2016). Requirements prioritization: challenges and techniques for quality software development. *Adv Comput Sci Int J*, 5(2), 14-21.
[Google Scholar](#)

- [93] Vijayan, V. K., Bindu, K. R., & Parameswaran, L. (2017, September). A comprehensive study of text classification algorithms. *International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 1109-1113. IEEE.
[Google Scholar](#) [CrossRef/DOI](#)
- [94] Awad, W. A., & ELseuofi, S. M. (2011). Machine Learning methods for E-mail Classification. *International Journal of Computer Applications*, 16(1), 39-45.
[Google Scholar](#)
- [95] 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), 100001.
[Google Scholar](#) [CrossRef/DOI](#)
- [96] Weng, W. H., Wagholikar, K. B., McCray, A. T., Szolovits, P., & Chueh, H. C. (2017). Medical subdomain classification of clinical notes using a machine learning-based natural language processing approach. *BMC medical informatics and decision making*, 17(1), 1-13.
[Google Scholar](#) [CrossRef/DOI](#)
- [97] Todd, J., Richards, B., Vanstone, B. J., & Gepp, A. (2018). Text mining and automation for processing of patient referrals. *Applied clinical informatics*, 9(01), 232-237.
[Google Scholar](#)
- [98] Hujainah, F., Bakar, R. B. A., Abdulgaber, M. A., & Zamli, K. Z. (2018). Software requirements prioritisation: a systematic literature review on significance, stakeholders, techniques and challenges. *IEEE Access*, 6(1), 71497-71523.
[Google Scholar](#) [CrossRef/DOI](#)
- [99] Kabukye, J. K., De Keizer, N., & Cornet, R. (2020). Elicitation and prioritization of requirements for electronic health records for oncology in low resource settings: A concept mapping study. *International journal of medical informatics*, 135(1), 104055, 1-11.
[Google Scholar](#) [CrossRef/DOI](#)
- [100] Aithal, P. S., & Kumar, P. M. (2015). Applying SWOC analysis to an institution of higher education. *International Journal of Management, IT and Engineering*, 5(7), 231-247.
[Google Scholar](#)
- [101] Pai, Y. (2021). Open Source Intelligence and its Applications in Next Generation Cyber Security- A Literature Review. *International Journal of Applied Engineering and Management Letters (IJAEML)*, 5(2), 1-25.
[Google Scholar](#) [CrossRef/DOI](#)
