Proceedings of the Second International Conference on Electronics and Sustainable Communication Systems (ICESC-2021)
IEEE Xplore Part Number: CFP21V66-ART; ISBN: 978-1-6654-2867-5

2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC) | 978-1-6654-2867-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICESC51422.2021.9532736

A study on Quality Assessment of Requirement Engineering Document using Text Classification Technique.

Shilpi Singh[1], L P Saikia[2], Sunandan Baruah[1]

Email: singhshilpi2410@gmail.com, lp_saikia@yahoo.co.in, sunandan.baruah@adtu.in

[1]Assam down town University, Panikheti, Guwahati Assam - 781026, India

[2]Girijanand Chowdhury Institute of Management and Technology, Azara Guwahati - 781017, Assam -India

Abstract: The Software Requirement Engineering document is the most important artifacts of the software development life cycle model. In majority of software systems, the Requirements Engineering (RE) Document or SRS (Software Requirement Specification) document has been **written in natural language English that are prone** to ambiguity. The ambiguous Requirement Engineering document may lead to disastrous results thereby hampering the entire development process and ending up compromising on the quality of a system. The success of any software product depends upon the quality of the Requirement Engineering document. The main reason for software crisis in software Industry is the Ambiguous Requirement Engineering Document. This paper discusses about the types of ambiguity, approaches to handle and providing a level of automatic assistance in order to detect ambiguity in the Requirement Engineering Document. The study also confirms the use of text classification technique to classify a text as "ambiguous" or "Unambiguous" at the syntax level. The key objectives of the work include understanding the presence of ambiguity in any Requirement Engineering document with the help of Machine Learning Techniques and finally minimizing or reducing it.

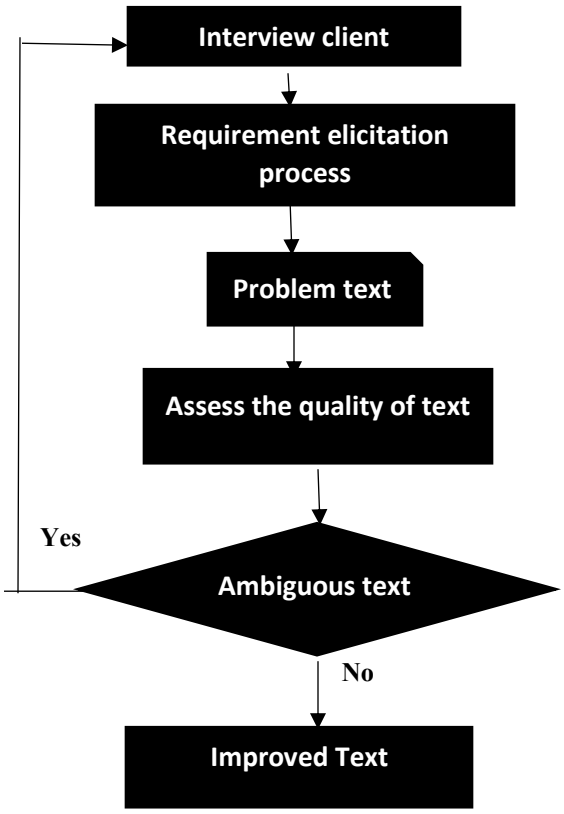
Keywords: Textual Ambiguity, Text Classification, Kappa index, Requirement Engineering (RE) Document.

1. INTRODUCTION

The important aspect of SDLC is Requirement Engineering, where the toughest part is to conceive the precise requirement of any software system before building it. The clients belongs from different background and thus, may sometimes introduces ambiguity in any Requirement Engineering (RE) document. It is mainly concerned with gathering, analyzing, and specifying and verifying user's requirement that is usually written in English and thus is able to induce ambiguity. The main objective of requirement analysis and specification phase is to specify the exact requirement of the system into Software Requirement Engineering Document. The quality of any text depends on how precisely and concisely the document is jotting down the requirements. Since the requirements specified by, the Clients and Users who are completely from different background and

have inadequate knowledge to specify the requirement unambiguously. The stages of identification of the problem is summarized in the Figure1. After interviewing the clients, the Requirement Engineering document is compiled and by assessing the text, we can improve the quality of the text. The previous work published in this area addressed the issue of detecting ambiguities and proposed many approaches. The most popular method to detect ambiguity is Inspection Techniques proposed by Bertrand Meyer [1, 2]. The manual approach considered the most accurate one but also the most expensive approach. The approach proposed by L Mich *et al* [3] focusses on semi-automated approach and using some NLP tool. In some approaches the researchers restricts the use of techniques developed in Natural Language Processing N. Kiyavitskaya et al [4]. The approach proposed by I Hussain [5] focusses on targeting textual ambiguity at the both conceptual and surface level. The conceptual level focusses on semantic understanding and thus requires deeper semantic knowledge. It was studied by understanding Meyer's Seven Sins [1]. The conceptual level understanding can, be studied by identifying the following identified features in any Requirement [2]. (i)Noise ii) Silence iii) Over – Specification iv) Contradiction v) Ambiguity vi) Forward Reference vii) Wishful Thinking. The surface level understanding mainly focusses on the syntax of the textual document and thus is easy to achieve [2 [3]. As discussed, researchers have previously attempted to study textual ambiguity and provided various manual, semi-automated and automated techniques to deal with textual ambiguity in the RE documents [4].

Figure 1. Flowchart for the proposed work



2. AMBIGUITY IN RE DOCUMENT

The ambiguity is defined as the multiple interpretations of a single statement. The ambiguity in the textual document is the most complicated scenario that occurs in literature, which is not good for the software development process. The lack of consistencies and completeness can be targeted using formal specifications but ambiguities in informal specifications is difficult to locate. The ambiguity in a textual document can be studied at the two levels:

Surface level – The surface understanding mainly focusses to exactly understand the concept stated in the textual document. It mainly focusses on the grammatical arrangement of any text. The list of attributes at the reading comprehension had been taken out from syntactic parser that can induce ambiguity to a text. [5]. The syntactic parser identified the following features that able to induce ambiguity in the textual data. The important attributes for the literal understanding are the number of words, adjectives, adverbs and conjunctions etc in a sentence.

b) Conceptual level – It mainly deals with the meaning of any text and requires deeper knowledge. It mainly targets the features identified by Meyers [1].

The Machine Learning is a branch of Artificial Intelligence, helps the system to automatically, learn from the experience without the tedious programming. The main intention is to allow the computer to learn automatically without human intervention.

3. QUALITY ASSESMENT OF THE TEXT

As shown in the Figure 1, the requirement given by the client is to be assessed by the help different techniques that we discussed in the section 1. The Inspection techniques are being used for improving the text quality. Further, the main objective of our proposed work is to build a system that can classify a document. By the proper training of the classifier, the text classification system will definitely benefit in assessing the quality of the text. As discussed in the section 1 of this paper, we can handle ambiguity using different manual, automated and semi-automated approach. In our study, we concluded that the semi-automated tools with the help of Machine Learning Techniques could be used to detect and resolve ambiguity from any RE document.

3.1 Machine Learning in Text Classification: The text classification is a technique that can classify a textual document into two or more than two categories based on the identified features. The text classification helped in detecting the subjectivity of the natural language. The Machine Learning techniques broadly classified as:

Supervised Machine Learning

i) Regression

ii) Classification

Unsupervised Machine Learning

i) Clustering

ii) Association

The ambiguity detection is a supervised Machine-learning problem in which can classify the RE document into two classes as “Ambiguous” or “Unambiguous” [4]. It will help the software developers to identify any ambiguity in the RE document before they could proceed for software development phase. A textual document can be classified as“Ambiguous” or “Unambiguous” based on the attributes identified.

3.2 Stages of text classification :

a) Document Preprocessing – The main steps involved in document processing are:

i) Tokenization of text- The tokenization process transforms tokens or words used in the process. The *tokenize class* in Java helps to break sequence of words into tokens.

ii) Stop word removal – The part of text like auxiliary verbs, articles and conjunctions that do not have much importance are deleted from a text in this process. These words stop words are removed from a text.

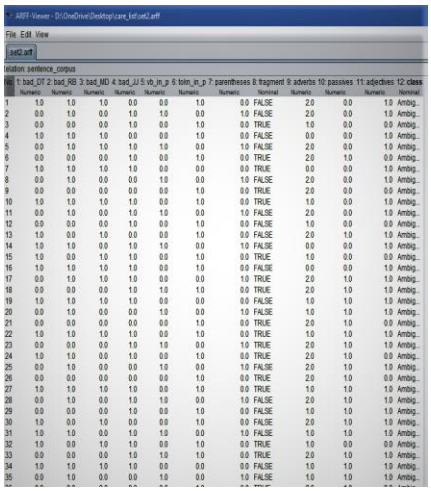
iii) Stemming words – The size of any text can be reduced by removing the words with the same root . The porter's algorithm implementation is used to perform this task

iv) Part of speech Tagging - The class (Ambiguous or Unambiguous) are targeted by observing the words in the text that has nouns, adjectives, adverbs, verbs etc. The Parts – of – speech are used to tag parts of speech.

v) Feature extraction and selection - The process of feature selection and extraction identifies only the main features can be targeted for classification [5]. The list of identified features for text classification listed in the Table 1. The features for classification are identified, by calculating the likelihood ratio calculated by annotators from different background [2, 6].

vi) Data representation in ARFF format The proposed research uses a very important and approves methodology called Bag of word for the task. The BOG is a method that simplifies the document representation by calculating the frequency of occurrence of word. The Figure 2 shows the sample dataset in ARFF format used for the analysis purpose. The dataset used is self-explanatory and can be used in the proposed research work.

Figure 2: Sample dataset in ARFF format using WEKA



The figure 2 shows a sample of the data set used for the classification purpose. The eleven identified features are included that can induce ambiguity to the textual document and the twelfth column

is the class "Ambiguous" or "Unambiguous" [6].

4. DECISION TREE BASED TEXT CLASSIFIER

The main task is to perform the automated quality evaluation of the textual document in terms of ambiguity by the help of text classification technique. The text classification is used to divide the contents or the documents into two categories. The table 1 shows the features that are used for the purpose of ambiguity detection. The presence of these attributes induces ambiguity in the textual document.The text classification is used to divide the documents into two categories on the basis identified features.

In our proposed work, we identified the features that to classify a text based on ambiguity present in any textual document [5]. The values of identifying features of the groups are collected from the available documents. In addition, when an unclassified document is supplied it can classify it based on the features identified.

Table 1: The Ambiguous features identified for classification

Sno.	Features	Description	Type of data
1	bad_DT	Bad determinat	Real
2	bad_RB	Bad Adverb	Real
3	bad_MD	Bad Modal	Real
4	bad_JJ	Bad Adjectives	Real
5	vb_in_p	Verbs in parenthesi	Real
6	Tokn_in_p	Token in parenthesis	Real
7	parenthesis	Parenthesis in a text	Real
8	Fragment	Fragments in a text	True / False
9	Adverb	Adverbs in a text	Real
10	Passives	Passive voice	Real
11	Adjectives	Adjectives in a text	Real
12	Class	The Binary class	Ambiguous or Unambiguous

3.1 Algorithm: In our problem of classifying a textual document, we choose the C4.5 a decision tree-learning algorithm since it allows backtracking and identified the reason for classification. The Java based Machine-learning tool used along with the framework of training and evaluation of the classifier.

The WEKA is a tool used for processing and visualization of data by implementing different Machine Learning Algorithms. The algorithm of our choice selected and used to run the dataset for the attributes or the features selected. It provides a statistical output of the model and visualization tool to inspect the data.

3.2 Prototype: A small instance in Java has been used to show the implementation of the classifier. The elucidated corpus designed by the I Hussain [4] used to train the text classification prototype that classifies the text at the surface level. The initial results confirms that the available NLP tools and the text classification technique can be very easily used to classify a text based on level of ambiguity in it.

Table 2: Result of using C4.5 algorithm for sentence classification

	Scheme	Correctly Classified Instance	Incorrectly classified Instance	Kappa
Corpus Size (472)	Training + Testing on same set	436 (92.37%)	18 (7.63%)	0.768
	Cross Validation (10 Folds)	418 (88.56%)	54 (11.44%)	0.65

5. EXPERIMENTAL ANALYSIS USING WEKA

The ambiguity detection dataset [2] simulated using the WEKA classifier on different classification algorithms like Naïve Bayes, Random tree and C4.5 algorithm. The RUN information generated using WEKA analyzed by comparing different statistical parameter [9,10]. The Table 2 summarized the result of implementing (Run Information) Naïve Bayes, Random tree and C4.5 (Implemented as J48 in WEKA) in the ambiguity detection dataset. The Table 2 indicates that the correctly classified and incorrectly classified instances for C4.5 algorithm (Implemented as J48 in WEKA). The classification accuracy for J48 Algorithm for Ambiguous class is 57.45% and for Unambiguous, class is 42.55%. The parameters considered for the J48 algorithm shows a substantial quality of the dataset and thus can be used, for text classification purpose [11, 12]. However, the classification accuracy is similar for the three algorithms. The precision value for J48 is better compare to other algorithms [6, 7]. The result summarized in the Table 3 indicates that the performance of J48 algorithm is comparatively better and thus, further uses the J48 algorithm for the prediction of the unseen dataset in order to classify a text as “ambiguous” or “Unambiguous”.

The C4.5 algorithm implemented in WEKA as J48 algorithm. The result of simulation performed on the ambiguity detection dataset indicates that the J48 algorithm used for text classification and it can further used to predict the unseen data [8].

Table 3: Simulation of various text classification Algorithm using WEKA

Algorithm	Precision	Recall	F Measure	Kappa Index	Accuracy
J48	0.913	0.778	0.840	0.768	57.45%
Random tree	0.786	0.812	0.800	0.518	42.53%
Naïve Bayes	0.834	0.830	0.834	0.656	41.51%

The classification accuracy for J48 Algorithm for Ambiguous class is 57.45% and for Unambiguous, class is 42.55%. The area under the ROC curve for J48 Algorithm is 0.8361, which indicates that a good quality dataset used for the classification purpose [9, 10]. The parameters considered for the J48 algorithm shows a substantial quality of the dataset and thus can be used, for text classification purpose [11, 12].

6. CONCLUSION AND FUTURE WORK

The ambiguities in a textual document is the major concern and the main reason for software crisis in development phase. The process of identification of ambiguous text will improve the quality of text and thus speed up the process of SDLC to save both time and cost. In order to prove our assumption of using text classification to classify a text as "Ambiguous" or "Unambiguous". The system has the performance accuracy of 80.65% by using 10 – fold cross validation techniques [13]. The experimental analysis performed by using WEKA indicates the substantial quality of data set for ambiguity detection. The study also confirms that the model can further be used for the prediction of instances of the unseen dataset [7]. The study also concluded that the text classification technique has the potential of detecting and resolving issues related to ambiguity in the Requirement Engineering text. The supervised learning and information retrieval strategies helps in improving the data set by underrating the text at the conceptual level [14, 15].

Our future work will mainly focus on detecting ambiguities at the conceptual level that requires deep semantic knowledge of the problem domain. Moreover, the classifier can be trained further to make it more robust and efficient.

7. REFERENCES / BIBLIOGRAPHY

[1] Meyer B, "On formalism in specifications", IEEE software, pp 6 – 20, 1985.

[2] Kamsties, E , Berry D M & Paech B, " Detecting Ambiguities in Requirement Engineering Document Using Inspection Techniques". In Proceedings of the First workshop on Inspection in Software Engineering (WISE'01), pp 68 – 70, Paris 2001.

[3] L Mich and R Garigaliano , " Ambiguity measure in Requirement Engineering ",In Proceedings of ICS , 16th IFIP , pp 39 -48, 2000.

[4] N. Kiyavitskaya , N Zeni , L Mich , and D M Berry, "Requirements for Tools for

Ambiguity identification and measurement in natural language specifications ," In Proceedings of WER'07,2007 , pp. 197, 2006.

[5] I Hussain, "Using text classification system to Automate Ambiguity Detection in SRS document", Master's Thesis, Computer science and Software Engineering Department, Concordia University, Montreal Canada, August 2007.

[6] Shilpi Singh and L P Saikia , "Feature extraction and performance measure of Requirement Engineering (RE) document using text classification technique", 4th Int'l Conference on Recent Advances in Information Technology, IEEE Explore , RAIT-2018 – IIT (ISM) – Dhanbad.

[7] Shilpi Singh and L P Saikia, "A Comparative Analysis of Text Classification Algorithms for Ambiguity Detection in Requirement Engineering Document using WEKA", 4th International Conference on ICT for Sustainable Development, SPRINGER – 2019.

[7] I Hooks, "Writing good requirements", Proc. of the fourth International conference in requirement engineering, Colorado International symposium of the NCOSE, Vol. 2, pp197-203,San Jose , CA ,1994.

[8] F. Fabbrini, M. Fusani, V. Gervasi, S. Gnesi, and S. Ruggieri, "Achieving Quality in Natural Language Requirements", Proceedings. 11th International Software Quality Week, San Francisco, 1998.

[9] F. Fabbrini, et al., "Achieving Quality in Natural Language Requirements," Proc. of 11th Int'l Software Quality Week Conference (QW'98), May 1998.

[10] M. Ikonomakis , S Kotsiantis , V. Tampakas , " Text Classification using Machine Learning Techniques", WSEAS TRANSACTIONS on COMPUTERS , Volume 4 , Issue 8 , pp. 966-974 , August 2005.

[11] Klein, D and Manning, C.D, "Accurate Lexicalized Parsing", proceedings of the 41st Meeting of the Association for Computational Liguistics,2003.

[12]. S. George and S. Joseph, "Text classification by augmenting Bag of Words (BOW) Representation with co-occurrence features", IOSR, pp. 34-38, June 2014.

[13]. S. K. Pandey and Mona Batra. Formal Methods in Requirements Phase of SDLC, I J C A (0975 – 8887) Volume 70– No.13, 2013.

[14]. Manoharan, Samuel. "Supervised Learning for Microclimatic parameter Estimation in a Greenhouse environment for productive Agronomics." Journal of Artificial Intelligence 2, no. 03, 2020.

[15]. Suma, V. "A Novel Information retrieval system for distributed cloud using Hybrid Deep Fuzzy Hashing Algorithm." JITDW 2, no. 03, pp151-160, 2020.