ISSN PRINT 2319 1775 Online 2320 7876

Research paper **Volume 11, Iss 12 , 2022**

AMBIGUITY DETECTIN TECHNIQUES OF SRS: CLASSIFICATION,MERITSANDDEMERITS

[1]M RRajaRamesh, [2]Prasad Devarasetty

[1]Department of CSSE, A U College of Engineering, Andhra University, Visakhapatnam,AndhraPradesh,India.

[2]Associate Professor,Dept. of CSE, DVR & DR HS MIC College of Technology,Kanchikacherla,AndhraPradesh,India.

Abstract

The success rate of a software project can be directly influenced by the Quality of Software Requirements Specification. There are different characteristics which affects the quality of SRS.As stated by IEEE Standard 830 there are eight characteristics (Unambiguity, Correctness, Completeness, Consistency, Ranked for importance and or stability, Modifiability, Verifiability and Traceability) which affect the Quality of SRS. Unambiguity is the one of the important characteristic which also influences the other characteristics like Correctness, Verification and Traceability of SRS. Out of eight characteristics stated by IEEE three characteristics are directly or indirectly affected by the ambiguity of the SRS. This indicates the importance of identification of ambiguity in the SRS documents. So far so many techniques were developed for detecting the same, but every technique has its own merits and demerits. This paper explores different approaches used for detecting the ambiguities in the SRS document and provides clear distinction among them.

Keywords: Ambiguity, Software requirements Specification, Verification, Correctness, Traceability

1. Introduction

Software development life cycle contains different phases like Analysis, Design Coding, Testing and Maintenance. Every phase produces an artifact which can be served as input to next following phase like analysis phase produces SRS document [11], and SRS can be served as input to Design Phase. Design phase produces technical design Document (TDD) which can be served as input to the coding phase. Coding phase produces the source code document (SCD) which can be served as an input to the Testing phase [27]. This flow of activities clearly indicates that if there is any mistake committed in the SRS document that can be propagated to design followed by coding and finally it may be caught in the testing phase. But if we find any defect after implementation or post implementation the cost of removal of that respective bug is 50 to 200 times more than the cost of removal of the same in the requirements engineering phase. It shows the significance of quality of SRS [9][10] document which directly impacts the remaining phases of Software development process and finally decides the quality and success rate of a software project.

SRS document can be used by almost all the members of respective Project development team. While using this document no two users can get the different meaning for same content [9][10].If the perception of a statement written in the requirements document varies from person to person then that particular requirement can be treated as ambiguous. If such ambiguous requirements are there in the SRS, that leads to critical bugs or even it causes the software to fail in performing the required respective functionality. To improve the quality SRS document, Technical design document and other documents Verification and Validation (V & V) activities [26] were introduced in1983.Verification means checking the process used

ISSN PRINT 2319 1775 Online 2320 7876
Research paper **© 2012 IJFANS. All Rights Reserved**, Volume 11, Iss 12 , 2022

to produce the product and Validation means checking the product produced after application of well-defined process. The main intension of V & V activities to concentrate on early testing; this indicates the quality verification of artifact produced at every stage of software development process. So, focusing on the quality of SRS document [11] in the requirements engineering phase leads to focus on characteristics of SRS which includes the one of the most important characteristic ambiguity [1]. In order to check the ambiguity of SRS in early days of software development, they used to rely on manual techniques like reviews and inspections [1][2][8]. After the invent Natural language processing techniques, slowly started using NLP techniques, later machine language processing techniques replaced the manual and NLP techniques. Even though NLP and ML techniques are better when compared with Manual techniques they cannot serve the complete purpose of identifying the ambiguity at all the levels. For detecting the ambiguity of SRS different techniques were developed in the past, which include manual approaches, approaches using Natural language Processing Techniques, and approaches using Machine learning techniques, but every technique has its own merits and demerits.

2. Classification of Ambiguity Detection Techniques

Ambiguity [1] detection techniques [9] are segregated based on the approach they have used, like Manual approaches [18], Natural Language Processing (NLP) based approaches[19][20][21] and Machine Language (ML) based approaches [22][23][24][2]. The following section briefly describes the foresaid approaches.

2.1 Manual approaches

2.1.1 Inspection: Inspection [1][2][18] is a manual technique which involves different activities like planning, preparation, crosschecking and record keeping, in which all the participants who are designated for inspection are requested to provide their understandability on each and every requirement stated in the Software requirements specification document[9]. After collecting the reports given by the inspectors, the team lead of the inspection compares all the reports with one another. If there are any differences in the understandability of requirements they are further inspected, at the end, if there is still difference of opinions on any requirements those particular requirements are treated as ambiguous requirements.

2.1.2 Reviewing: In this approach [1][2] Stakeholders havingextensive knowledgein preparingthe SRS documents are selected for reviewing the SRS for detecting its ambiguity. The reviewers manually read each and every requirement present in the SRS document and provide their comments or opinions. Reviewers also provide the severity level of the ambiguity present in the SRS[9]. Based on the feedback delivered by the reviewers the authors of SRS will recheck with the customers or clients and prepares another version of SRS.

Merits and demerits of Manual approaches:

Merits:

ISSN PRINT 2319 1775 Online 2320 7876
Research paper **© 2012 IJFANS. All Rights Reserved**, Volume 11, Iss 12 , 2022

1. Human intervention in document verification is useful for improvement.
2. Ambiguity at different levels can be verified.
3. Personal understanding leads to uncover improved areas.
4. Easy to understand and easy to apply.
5. Best suitable for Small scale Projects
6. Cost effective.

Demerits:

1. Human mistakes can be done while verifying manually.
2. Manual verification may take more time for large and complex projects.
3. Not suitable to Medium, Complex and Large scale projects.

2.2 NLP based approaches

2.2.1 Automated Requirements Measurement (ARM): This tool was developed in 1990s by NASA for analyzing and reporting the quality of requirements written in the SRS document. The report was grounded on statistical investigation of word occurrences at several structural levels of the SRS document. ARM [19] uses various quality indicators like Depth Specification, Structure of text, extent, imperatives, continuances, directives, Weak phrases and Options. For ambiguity identification ARM tool uses Weak phrases and Options, for this purpose it has identified twelve predefined weak phrases. ARM Tool mainly focused on word level ambiguity.

2.2.2 Quality Analyzer for Requirement Specifications (QuARS): QUARS Tool takes the SRS document as input and first performs lexical analysis followed by syntax analysis for identification of defects, and then it applies the QUARS quality model for ambiguity detection of SRS document. This tool GUI contains three frames like Input frame, Dictionary frame and Output frame. This tool requires the Input document must be in the plain text format. To detect the ambiguity in the SRS, QuARS [20] considers words used in the requirement statements and it also focuses on the statements. Sometimes ambiguity at word level may be cleared when it is crosschecked their usage with the sentence, for this purpose QuARS [20] considers both word level and Sentence level ambiguity. It uses different set of patterns for identifying the ambiguity in the requirements of SRS Document.

2.2.3 Requirements Template Analyzer (RETA): In this Tool, requirements represented in the natural language are compared with different predefined Templates. These templates are designed based on the experience in preparing software requirements documents and are used as standards or guidelines in preparing the SRS documents. RETA tool also supports Rupps and EARS Templates. This tool focused more on word level and less on Sentence level ambiguity. This tool mainly concentrates on word level ambiguity.

2.2.2 Requirements Complexity Measurement (RCM): RCM tool [21] used to measure the complexity of requirements stated in the SRS document. This tool contains three layers. First layer for requirements preprocessing, second layer for application of complexity metrics and third layer for results parser. The second layer of the RCM tool uses ten metrics for measuring the complexity of requirements represented in natural language. This tool does not measure the entire complexity of the requirements document, it measures only the selected

ISSN PRINT 2319 1775 Online 2320 7876
Research paper **© 2012 IJFANS. All Rights Reserved**, Volume 11, Iss 12 , 2022

aspects of the document for which complexity metrics are developed. It accepts only the documents which are in the Microsoft excel file format. In this tool [21] metrics are developed by taking the number of words used, number of conjunctions(for example: still, therefore, in addition to , in case, rather than, although, so that, etc) used, number of ambiguous phrases used, number of documents referenced and other language barriers that forces the statements or requirements to become ambiguous.

Merits and Demerits of NLP Approaches:

Merits:

1. Word level ambiguity can be identified effectively.
2. Accuracy increases with increased training data.
3. Very effective in finding word level ambiguity.
4. Time consumption is very less.
5. Applicable to Small, Medium and Large scale Projects.

Demerits:

1. Ease of understanding and application is bit difficult.
2. Minimal focus on sentence level ambiguity
3. Unable to find out domain level ambiguity.
4. For complex requirements results are not promising.

2.3 ML based approaches:

2.3.1 Support Vector Machine (SVM) text classification: SVM is a supervised machine learning approach for text classification. This is useful for classification of high dimensional data. SVM can be used for linear datasets and nonlinear datasets. After preprocessing of data, Training datasets and Test datasets are prepared and the SVM text classification technique is applied for ambiguity detection of SRS document. SVM text classification [23] completely based on word level ambiguity. All the words appeared in the SVM model are labeled with Parts of Speech and then weights are given to the corpus[10]. A weight given in the corpus is compared with threshold value. Support vector machine [23] identifies the faults, it is difficult to identify or if the weight is large.

2.3.2 Statistical Machine translation (SMT) using the n-gram model: It is a machine translation methodology that takes the huge volumes of multi-lingual data to discover the utmost probable translation for a certain input. SMT systems [24] study to translate by examining the statistical associations between original texts and their present human renditions.n-gram model [24] is a statistical language model. In the n-gram model n indicates the number of words in the sequence; it catches the probability distribution over word classifications, like other models this also depends on the training corpus. SMT using n-gram model [24] is completely based on Word level ambiguity. This model predicts the next word based on the previous words in the sentence. It considers n previous words for predicting the future word that is why it is called as n gram model.

ISSN PRINT 2319 1775 Online 2320 7876
Research paper **© 2012 IJFANS. All Rights Reserved**, Volume 11, Iss 12 , 2022

2.3.3 Decision-Tree Text Classification: Decision tree algorithm is a supervised learning algorithm. In decision tree algorithm, Tree data structure is used to forecast the outcome of the required problem. It follows a top-down approach. These are built on divide and conquer approach. Decision trees divide the given input data into thick regions and thin regions. The binary tree splitting may be either binary or multiway. The algorithm splits the data until all the data becomes sufficiently homogeneous. After completion of training the output of decision Tree can be used for optimal classified predictions.Decision trees [22] contain two different types of nodes Decision nodes and Leaf nodes. Leaf nodes are results of decision taken by the decision nodes. Based on the decisions taken the size of the tree grows. As part of the search process, each node endures an assessment between numerous, implausible probable values. Decision tree Text classification [22] also considers word level ambiguity only.

2.3.4 Naive Bayes (NB) text classification: Naive Bayes algorithm [25] is also a supervised learning algorithm grounded on Bayes theorem and used for resolving classification problems. It is mostly used for text classification (Sentiment analysis) which contains high dimensional training dataset. It adopts that predictors in this model are tentatively autonomous, or unconnected to other features in the Naïve Bayes model. This model assumes that the outcome is based on equal contribution of all the features. This classifier can be applied for making quick predictions. Its predictions are based on the probability of the objects. The NB text classifier [25] is trained depending on the word possibility and reckoning. To classify a text, it examines the probabilities of words erudite through training data.

Merits and Demerits of ML based approaches:

Merits:

1. More powerful and can process large and complex datasets.
2. Decision Tree like models is easy to interpret and implement when compared with other ML based models.
3. Applicable to Small, Medium and Large scale Projects.
4. Very effective in finding word level ambiguity.
5. Less cost

Demerits:

1. Ease of understanding and application is bit difficult.
2. Less focus on Sentence level ambiguity.
3. Domain level ambiguity is not considered.
4. Training of the model is time consuming.
5. Difficult and it can be tough to know what patterns they depend on.
6. No models are 100% perfect.

3. ComparisonandAnalysis:

In this section the ambiguity detection approaches for SRS document are compared against different attributes like their ease of learning, ease of application, applicability to Complex and large scale projects, applicability to medium size projects and applicability to small scale

ISSN PRINT 2319 1775 Online 2320 7876
Research paper **© 2012 IJFANS. All Rights Reserved**, Volume 11, Iss 12 , 2022

projects along with this the aforesaid approaches are also compared based on the level of detection of ambiguity i.e ambiguity at word level, ambiguity at sentence level and ambiguity at domain level. Ambiguity at word level means words used in requirements statements have multiple meanings. Sentence level ambiguity means sentences used to write the user stories may have multiple meanings. Domain level ambiguity means meanings of few words may vary with the domain. After survey and analysis our observation is different approachesaretravailwithdifferentproblemsandnoapproachisshowingcompletesatisfactioninall aspects. This clearly indicates that rigorous research in ambiguity detection of SRS is required so that ambiguity at all the levels can be identified for the sake of producing quality Software. The following tables represent the comparison of ambiguity detection approaches against different attributes.

Table1: Differentambiguity detection approaches and their applicability

S.No	Approach	1	2	3	4	5
1	Manual Approaches	√	√	√	×	×
2	NLP based Approaches	×	×	√	√	√
3	ML Based Approaches	×	×	√	√	√

1:Ease of Learning
2:Ease of Application
3: Applicability on Small scale Projects
4:Applicability on Medium scale Projects
5: Applicability onComplex and Large scale Projects

Table-2: Differentambiguity detection approaches and their level of detection of Ambiguity.

S.No	Approach	1	2	3
1	Manual Approaches	√	√	√
2	NLP based Approaches	√	√	×
3	ML Based Approaches	√	×	×

1: Ambiguity at Word level
2: Ambiguity at Sentence Level
3: Ambiguity at Domain level

4. Conclusion:

Out of eight good characteristics of SRS(as mentioned by IEEE 830), Unambiguity is the most important characteristic which requires a special attention. The intension of ambiguity detection in the SRS document leads to preparing a path for achieving the quality of SRS document. After examining the different techniques available in the literature, the important

ISSN PRINT 2319 1775 Online 2320 7876
Research paper **© 2012 IJFANS. All Rights Reserved**, Volume 11, Iss 12 , 2022

observation is that most of the techniques whether NLP based approaches or ML based approaches are trying to find out word level ambiguity and very few are focusing very little on sentence level ambiguity. But no NLP or ML based approach is considering the domain level ambiguity which is also most important. Even though manual approaches are focusing on the Sentence level and domain level ambiguity, their application is very limited as they are difficult and time consuming in handling large and complex projects. So, there is a need to concentrate on creating a new model which is capable of detecting ambiguity at all the three levels like Word, Sentence and Domain levels.

5. References

[1] Bussel DV. Detecting ambiguity in requirements specifications. 2009(09).

[2] Oo, Khin&Nordin, Azlin& Ismail, Amelia Ritahani&Sulaiman, Suriani. (2018). An Analysis of Ambiguity Detection Techniques for Software Requirements Specification (SRS).IJET,7.501.10.14419/ijet.v7i2.29.13808.

[3] Yang H, Willis A, Roeck AD, Nuseibeh B, De Roeck A. Automatic detection of nocuous coordination ambiguities in natural language requirements. Proceedings of the IEEE/ACM international conference on automated software engineering SE - ASE '10. 2010:53-62.

[4] Misra J, Das S. Entity Disambiguation in Natural Language Text Requirements. 2013 20th Asia-Pacific Software Engineering Conference (APSEC). 2013:239-46.

[5] Polpinij J. An ontology-based text processing approach for simplifying ambiguity of requirement specifications. 2009 IEEE Asia-Pacific Services Computing Conference, APSCC 2009. 2009:219-26.

[6] Maroulis G. Comparison between Maximum Entropy and Naïve Bayes classifiers: Case study ; Appliance of Machine Learning Algorithms to an Odesk ' s Corporation Dataset GeorgiosMaroulis Submitted in partial fulfilment of the requirements of Edinburgh Napier University. 2014(January).

[7] Brown PF, DeSouza PV, Mercer RL, Della Pietra VJ, Lai JC. Class-Based n-gram Models of Natural Language. Computational Linquistics. 1992;18(1950):467-79.

[8] Tyler Baldwin YLBAIRS. Automatic Term Ambiguity Detection. Acl. 2013;2:804-9.

[9] M.R Rajaramesh, et al, Metrics for Software Requirements Specification Quality Quantification",International Journal of Computers and Electrical Engineering (IJCEE), ELSEVIER,Oct 2021.

[10] R Raja Ramesh Merugu, et al, Automated Cloud service based quality requirement classification for software requirement specification, Evolutionary Intelligence, Springer-Verlag GmbHGermany, SPRINGER, May 2019.

[11] M.R Raja Ramesh, et al. Activity based Quality Assessment Technique for Software Requirement Specification, International Journal of Innovative Technology and ExploringEngineering (IJITEE) ISSN: 2278-3075, Volume 8 Issue-2S2, December, 2018.

[12] M.R Raja Ramesh, et al .A Survey on Security Requirement Elicitation Methods: Classification, Merits and Demerits", International Journal of Applied Engineering Research ISSN 0973-4562 ,Volume 11, Number 1 pp 64-70, jan-2016.

[13] M. Ceccato, N. Kiyavitskaya, N. Zeni, L. Mich, D. M. Berry, Ambiguity identification and measurement in natural language texts, Technical Report, University of Trento, 2004.

[14] A. O. J. SABRIYE, W. M. N. W. ZAINON, An approach for detecting syntax and syntactic ambiguity in software requirement specification., journal of theoretical & applied information technology 96 (2018).

ISSN PRINT 2319 1775 Online 2320 7876

Research paper **© 2012 IJFANS. All Rights Reserved**, Volume 11, Iss 12 , 2022

[15] N. Kiyavitskaya, N. Zeni, L. Mich, D. M. Berry, Requirements for tools for ambiguity identification and measurement in natural language requirements specfications, requirements engineering ,13 (2008) 207-239.

[16] PreranaChaithra, Dr. ShantharamNayak,"Machine Learning Technique for Identifying Ambiguities of in SoftwareRequirements" Turkish Journal of Computer and MathematicsEducation(TURCOMAT), Vol.12, No. 11,2021, pp 6852-6857.

[17] PreranaChaithra, ShantharamNayak," Quality Assurance Techniques in SRS Documents", Elsevier-Scopus indexed International Journal of Innovative Technology and Exploring Engineering (IJITEE), ISSN:2278-3075, Vol-9, Issue- 2S, December 2019, pp. 14-18.

[18] Kamsties E., Berry D.M. and Paech B., 2001. Detecting Ambiguities in Requirements Documents Using Inspections, Proceedings of the First Workshop on Inspection in Software Engineering.

[19] Nathan Carlson,etal.The NASA automated requirements measurement tool: a reconstruction, Innovations in Systems and Software EngineeringVolume 10Issue 2June 2014pp 77–1https://doi.org/10.1007/s11334-013-0225-8.

[20] StefaniaGnesi and GianlucaTrentanni, QuARS A NLP Tool for Requirements Analysis,Sept 2005,CMU/SEI-2005-TR-014,ESC-TR-2005-014.

[21] V. Antinyan, M. Staron, A. Sandberg and J. Hansson, "A Complexity Measure for Textual Requirements," 2016 Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement (IWSM-MENSURA), Berlin, Germany, 2016, pp. 148-158,doi:10.1109/IWSM-Mensura.2016.030.

[22] Ormandjieva O, Kosseim L. Automatic quality assessment of SRS text by means of a decision-tree-based text classifier. Proceedings – International Conference on Quality Software. 2007(Qsic):209-18.

[23] Xu, Z., Yu, K., Tresp, V., Xu, X., Wang, J. (2003). Representative Sampling for Text Classification Using Support Vector Machines. In: Sebastiani, F. (eds) Advances in Information Retrieval. ECIR 2003. Lecture Notes in Computer Science, vol 2633. Springer,Berlin,Heidelberg.https://doi.org/10.1007/3-540-36618-0_28.

[24] Gong et al., N-gram-based Tense Models for Statistical Machine Translation, https://aclanthology.org/D12-1026,EMNLP 2012.

[25] Xu,S. (2018). Bayesian Naïve Bayes classifiers to text classification. Journal of Information Science,44(1),48–59. https://doi.org/10.1177/0165551516677946.

[26] Ciftcioglu, Ö. (1994). Verification and Validation in the Life-Cycle of Real-Time Software Development. In: Halang, W.A., Stoyenko, A.D. (eds) Real Time Computing. NATO ASI Series, vol 127. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-88049-0_101

[27] D. Ganesh, J. R Kumar, K. K Rao, Empirical Investigations to Find Illegal and its Equivalent Test Cases using RANDOM-DELPHI, IJSEA, Vol. No 11,pp 107-116, 2015.