Turkish Online Journal of Qualitative Inquiry (TOJQI)
Volume 12, Issue 7, July 2021: 8572 - 8577

Research Article

Minimization of Ambiguities in Software Requirements Specification

Prerana Chaithra[1], Dr. Shantharam Nayak[2]

[1]VTU Research Scholar, RVCE & Associate Professor, Information Science & Engineering, Sapthagiri College of Engineering, affiliated to VTU, Bengaluru

[2]Professor & Principal advisor, affiliated to Visveswaraya Technological University, Bengaluru

[1]preranachaithra15@gmail.com, [2]shantaram_nayak@yahoo.com

Abstract:

The most important aspects for software development are the software requirements. They are the foundation stone for initiating any software development process. Software requirements documents contain the needs of the customers in natural language. The content of the software requirement can be checked manually by using various methods like reviews, inspections and walkthroughs. In recent years there is an attempt to automate these activities as a result of advancement in automation of natural language analysis. Automation of text mining techniques and text analysis is leading to feasibility of automation of requirements documents processing. The process can be completed in minutes now which were taking weeks earlier. Automation of analysis of text has triggered numerous possibilities for quality assurance of requirements. The possibilities of automation are model checking automation, automated rule checking, automated test case execution and measurement automation. In future more tools will enter the scene for automation of requirements quality assurance. At present most of them are in experimental stage. There is a definite need for more research on how to measure and quality assurance of requirements.

Keywords: Walkthroughs, Software Requirement Specification, Text Mining, Ambiguity, Natural Language Processing, Quality Assurance

1. Introduction

In a software project development, requirements engineering is significant and effortful task. Usually software requirements are stated in natural language. There is a possibility that natural language requirements may contain ambiguous and incomplete details [2]. There is a need for method for identification where in quality requirements of software requirements are stated and which characteristic class the requirement belongs to. If it is possible to implement such requirements, it is easy to know how much quality requirements are defined against functional requirements and their distribution across characteristic categories like compatibility, performance, usability, security and reliability. It is also possible to understand how structured they are in specification document [1].

To document software requirements, SRS is used that contain both non-functional and functional requirements. This document provides basis for next stages of project development such as planning,

design, coding and testing [3]. Analysis of specification is a challenging task because the requirements document converts the real world problems into computer models that are verifiable [4]. As per the research [5], the reason for software errors are mainly due to defects in requirements i.e. up to 85%. The defects are to be removed in early stage of development; otherwise the cost of fixing the errors in later stages would increase intensely.

2. Literature Survey

There have been numerous attempts to identify and analyze the ambiguities encountered in a document. Some the notable efforts worth indicating involves development of tools for different applications pertaining to various domains[8][9][10][11]. Also, over a decade and half, significant attempts have been made to identify the analyze the ambiguities in requirement specifications to facilitate their automatic detection[12][13][14][15][17]. However, the attempts can be further extended to obtain reliable detection tools for SRS documents, assisting in easy detection of ambiguities and improvement of the document quality.

The present effort aims at developing a tool which not only identifies the ambiguity, but also explains the lines on which it can be improved. The tool indicates the line number where the ambiguity exists and the level of ambiguity listed as per the ambiguity datahandbook[1]. It also identifies the synonyms for the nouns and verbs, facilitated by parts-of-speech (POS) tagging, which assists in finding the right word to eliminate ambiguity in the sentence. After each cycle of processing, the percentage of ambiguity is calculated thereby quantifying the ambiguity existing in the document. This also enables to concentrate on those areas in the document where the ambiguity has serious consequences on the quality of the document. It also saves time taken for eliminating the ambiguity by identifying the right sentences or paragraphs.

3. Methodology

A SRS Tool is developed which helps to identify, analyze and reduces ambiguities in the software requirements. The model takes SRS documents as input. These documents are pre-processed. The pre-processing work includes task like extracting boundary sentences from the SRS document and replacement of images, tables and figures in the correspondent written format. The model is trained and tested for accuracy.

The proposed methodology is as follows.

Step 1: Start
Step 2: SRS document is input.
Step 3: Tokenization of the SRS document.
Step 4: lemmatization of the document is done.
Step 5: If ambiguity is present then detect ambiguity
Step 6: Analyse ambiguity
Step 7: Suggest modifications
Step 8: Incorporation of suggested modifications.
Step 9: Unambiguous SRS is obtained

Step 10: End

The proposed model is based on machine learning. The machine learning algorithm takes SRS document as input and provides output with ambiguous words highlighted. The model training has been done using a huge database of SRS documents

4. Results

The SRS document is a document that describes all the externally observable behaviours and characteristics expected of a software system. A Quality SRS is one that contributed to successful, cost effective creation of software that meets users' requirements. Some of the screenshots are shown in the following figures.

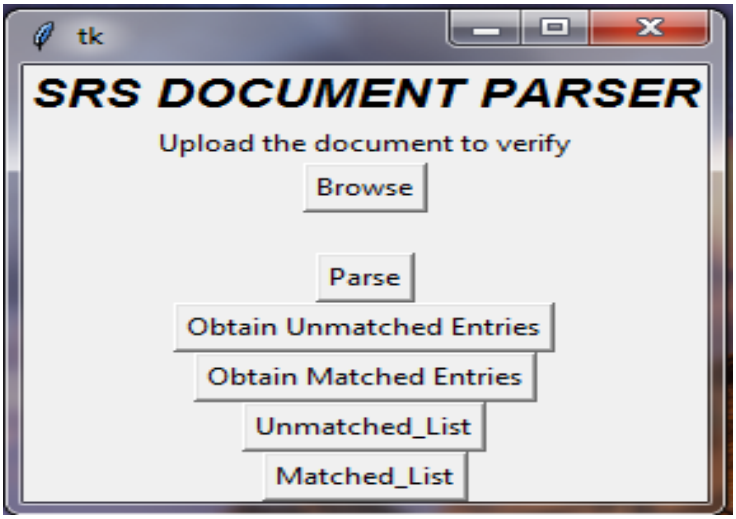

Figure 1 : SRS Document Parser [18]

The opening screen shot of the prototype application is as shown in figure 1. The screen has options to choose the file (SRS doc) and check the options like matched and unmatched entries.

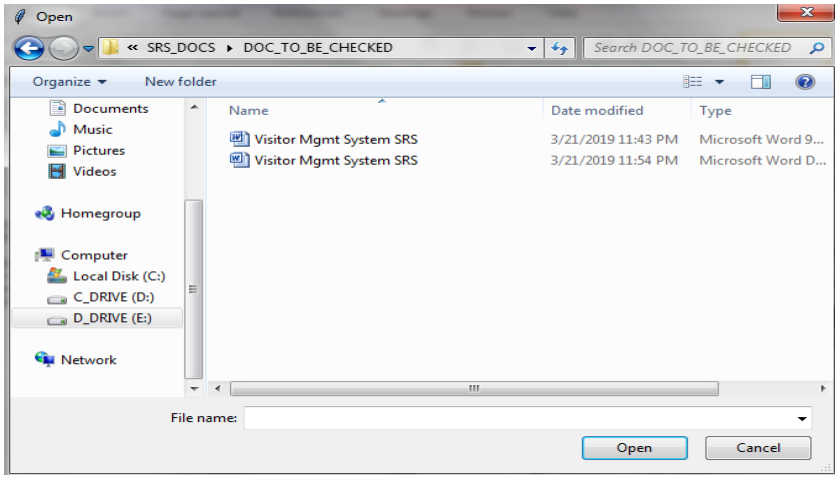

Figure 2: Selection of SRS document[18]

The selection the SRS document file is depicted in figure 2.

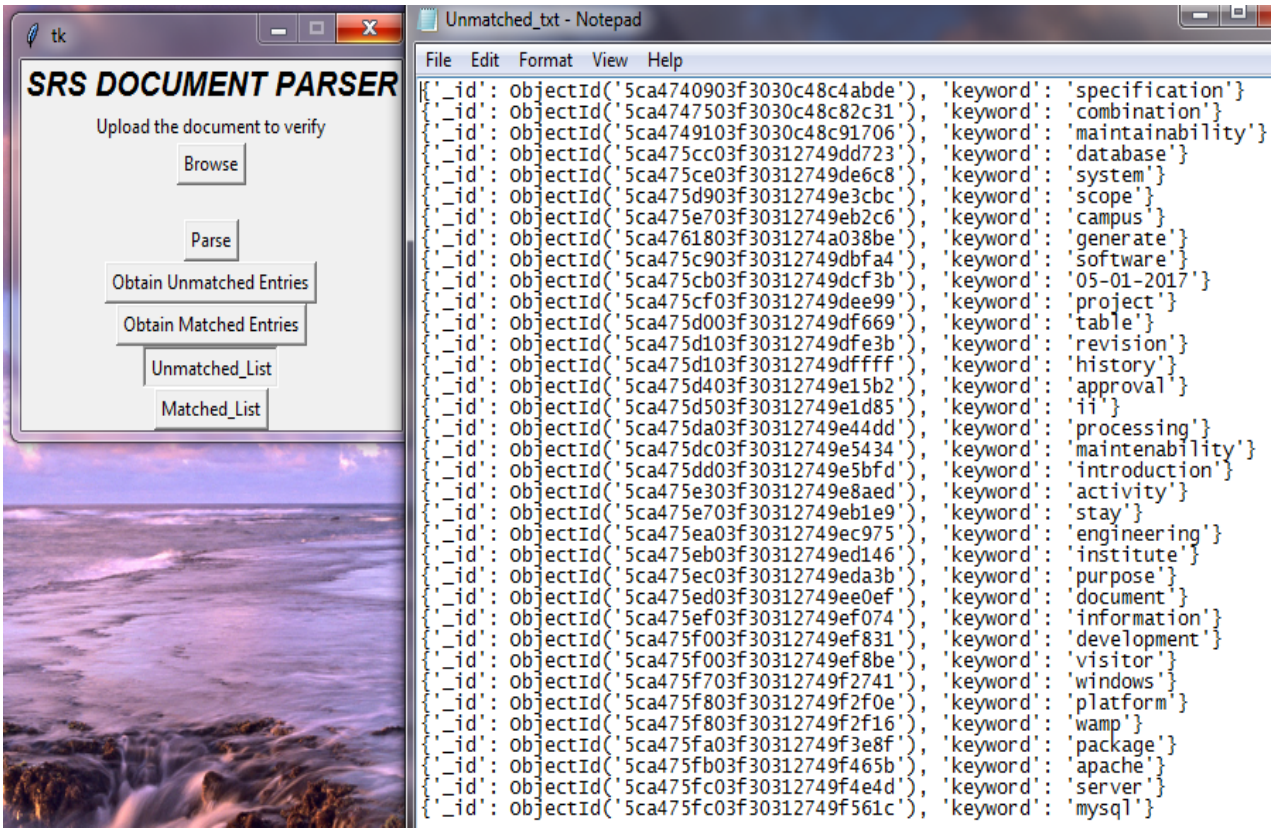

Figure 3 : Parsing process

The parsing process is as shown in the figure 3. The process searches for the keywords and extracts the matched and unmatched entries.

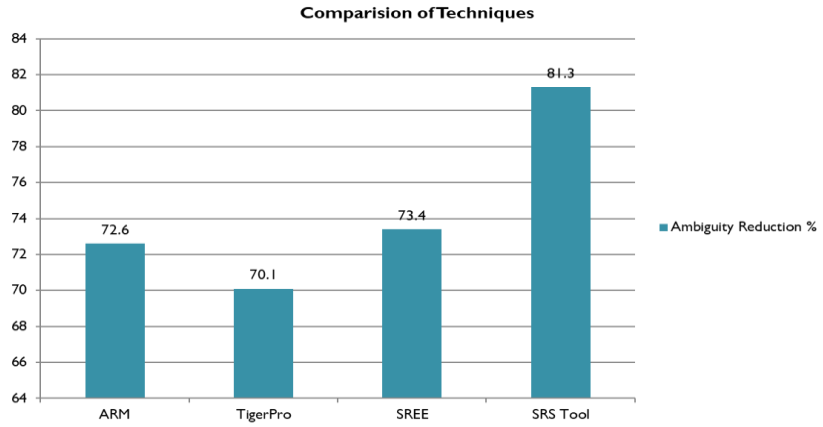

Figure 4 : Comparison of the available disambiguation tools

The Comparison of the available disambiguation tools is as shown in the figure 6. This shows that the proposed tool is better than the currently existing tools for reducing ambiguity in software requirements.

5. Conclusion

A tool is developed to detect, analyze and minimize ambiguities in the software requirements. SRS document is taken as input, pre_processed and ambiguities are detected and analysed by this tool. Later the tool helps in minimizing the ambiguities. Thus, unambiguous SRS is obtained. Using this SRS software can be developed. This leads to the development of high quality software which satisfies the customer needs. This tool is of very good accuracy of around 81 percent. Thus, making it better than many available disambiguating tools.

References:

[1] D. M. Berry, E. Kamsties, M. M. Krieger, From contract drafting to software specification: Linguistic sources of ambiguity, Technical Report, Technical Report, School of Computer Science, University of Waterloo . . . , 2003.

[2] L. Mich, M. Franch, P. L. Novi Inverardi, Market research for requirements analysis using linguistic tools, Requirements Engineering 9 (2004) 151-151.

[3] 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).

[4] 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.

[5] B. Gleich, O. Creighton, L. Kof, Ambiguity detection: Towards a tool explaining ambiguity sources, in: International Working Conference on Requirements Engineering: Foundation for Software Quality, Springer, pp. 218-232.

[6] B. K. Britton, Lexical ambiguity of words used in english text, Behavior research methods & Instrumentation 10 (1978) 1-7.

[7] 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.

[8] R. Garigliano, D. Nettleton, Neo-pragmatism, The LOLITA Project: the First Ten Years 3 (1997).

[9] R. G. Morgan, R. Garigliano, P. Callaghan, S. Poria, M. H. Smith, A. Urbanowicz, R. Collingham, M. Costantino, C. Cooper, L. Group, et al., University of durham: description of the lolita system as used in muc-6., in: Sixth Message Understanding Conference (MUC-6): Proceedings of a Conference Held in Columbia, Maryland, November 6-8, 1995.

[10] R. Garigliano, A. Urbanowicz, D. J. Nettleton, Description of the Lolita system as used in muc-7, in: Proceedings of the 7th Message Understanding Conference (MUC{7), Citeseer.

[11] L. Mich, R. Garigliano, Nl-oops: A requirements analysis tool based on natural language processing, WIT Transactions on Information and Communication Technologies 28 (2002).

[12] Prerana Chaithra, Shantharam Nayak, " 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

[13] L. Mich, R. Garigliano, et al., Ambiguity measures in requirement engineering, in: International Conference on Software Theory and Practice. ICS.

[14] F. Fabbrini, M. Fusani, S. Gnesi, G. Lami, The linguistic approach to the natural language requirements quality: benefit of the use of an automatic tool, in: Proceedings 26th Annual NASA Goddard Software Engineering Workshop, IEEE, pp. 97-105.

[15] W. M. Wilson, L. H. Rosenberg, L. E. Hyatt, Automated analysis of requirement specifications, in: Proceedings of the 19th international conference on Software engineering, pp. 161-171.

[16] Prerana Chaithra, Surekha R., Shantharam Nayak "Influence of Mathematics on Software Engineering", International Journal of Management Technology and Engineering, , ISSN: 2249-7455, Vol-9, Issue-5, May 2019, pp.3541-3547

[17] M. Luisa, On the use of ambiguity measures in requirements analysis", in: Applications of Natural Language to Information Systems, 6th international workshop NLDB'01, Gesellschaft für Informatik e. V.

[18] Prerana Chaithra, Dr. Shantharam Nayak, "Machine Learning Technique for Identifying Ambiguities of in Software Requirements" Turkish Journal of Computer and Mathematics Education(TURCOMAT), Vol.12, No. 11 2021, pp 6852-6857