Volume 5, Issue 10, October-2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Ambiguity in Requirement Engineering Documents: Importance, Approaches to Measure and Detect, Challenges and Future Scope

Shilpi Singh
Assistant Professor
Department of Computer Science & IT
Lalit Chandra Bharali College
Guwahati, Assam, India

Dr. Lakshmi P. Saikia
Professor
Department of Computer Science & Engineering
Assam Downtown University
Guwahati, Assam, India

Abstract: One of the important phases of software development life cycle (SDLC) is requirement engineering. Since it is very important to understand the exact requirement of any system before actually building it. But RE document is always written in natural language and which is prone to ambiguity. Thus it is very important to measure and detect ambiguity in RE document to decrease the cost and time ofsoftware development. This paper mainly discusses about the sources and reasons for of ambiguity and approaches used to avoid, detect or measure ambiguity in any RE text.

Keywords: Software engineering, software crisis,Requirement engineering text (RE text), GOORE (Goal oriented requirement engineering), ontology , lexical ambiguity , pragmatic ambiguity , Syntactic ambiguity , Natural language processing and tools .

I. INTRODUCTION

The main goal of requirement analysis and specification phase is to clearly understand the users requirement and to systematically organize the requirement into a specification document.A study by the Standish Group shows that the USA spends "$250 " billion each year on IT application development of approximately 175,000 projects and 31.1% of those getting cancelled before they ever get completed and this situation is termed as *software crisis* in software industry. Moreover, software purchases as compare to that of hardware purchases have been showing a worrying trend over the years[1].The achievement of the quality of software requirement is the first step towards software quality. Organizations are spending larger and larger portions of their budget on software compare to hardware. The software products are also hosts of some other problems since they are difficult to alter, debug and enhance users resources non optimally often fails to meet the users requirement. The study then reveals that "*Incomplete requirements and specifications* " are the primary reason for their failure and suggests "*CLEAR*" requirement statement as one of the key factors of successful software development[2,3].

RE is mainly concerned with gathering, analyzing, specifying and validating users requirement that are mostly documented using natural language. Despite the inherent ambiguity and informally that determines difficulty in proving the correctness, the natural language is widely used in software industry for specifying RE text. The IEEE standards mentioned that the ambiguous nature of the natural language makes the RE document ambiguous and thus degrades the overall quality of the RE document and they are called as "MEYER'S SEVEN SINS " and are as follows:

a) Noise
b) Silence
c) Over - specification
d) Contradiction
e) Ambiguity
f) Wishful Thinking
g) Forward reference [4]

The main measure of the success of a software system is the degree to which it meets its purpose. Therefore, identifying this purpose must be one of the main activities in the development of software systems. The practices the software industries are following for specifying the RE text itself has some loop holes other than the difficulties introduced by natural language and thus those practices of software industry need to be rectified. But still the use of natural language for specifying requirements have many advantages .It has been recognized that inadequate, incomplete, ambiguous, or inconsistent requirements have a significant impact on the quality of software. Thus, Requirements Engineering (RE), a branch of software engineering that deals with elicitation, refinement, analysis, etc. of software systems requirements gained a lot of attention in the academia as well as in the industry.

II. AMBIGUITY IN RE TEXT

Requirement engineering identifies the purpose and properties of a software system. It creates document in a form that is suitable to analysis, communication, and subsequent implementations . There are many problems occurring in the field of requirement engineering and are as follows:

- Requirements are not always obvious and have many sources.
- Requirement are not always easy to express clearly in words.
- Requirements are not always complete.
- Requirements are not always unique.
- Requirements are not always consistent.
- Requirements are related to one another.
- Requirements change.
- Requirements are neither equally important nor easy to meet.
- Requirement can be time sensitive.
- The number of requirements become unmanageable.

The requirements are the foundation for delivering quality software. It is often found that the short development cycle leads to cut short the time they will spend on the requirement analysis.One of the basic and oldest definitions of Requirements Engineering states that *"requirements definition is a careful assessment of the need that a system is to fulfill. It must say why a system is needed, based on current or foreseen conditions, which may be internal operations or external market. It must say what system features will serve and satisfy this context. And it must say how the system is to be constructed"*.

Thus any requirement engineering document need to be correct , consistent and unambiguous since it is the document of communication between the customers and the clients .

The major steps in the process of requirement engineering involves :

- *Domain analysis*: the environment for the system-to-be is studied. The relevant stakeholders are identified and interviewed. Problems with the current system are discovered and opportunities for improvement are investigated. Objectives for the target system are identified.
- *Elicitation*: alternative models for the target system are analyzed to meet the identified objectives. Requirements and assumptions on components of such models are identified. Scenarios could be involved to help in the elicitation process.
- *Negotiation and agreement*: alternative requirements and assumptions are evaluated. Risks are analyzed by the stakeholders and the best alternatives are selected.
- *Specification*: requirements and assumptions are formulated precisely.
- *Specification analysis*: the specifications are checked for problems such as incompleteness, inconsistency, etc. and for feasibility.
- *Documentation*: various decisions made during the requirements engineering process are documented together with the underlying rationale and assumptions.
- *Evolution*: requirements are modified to accommodate corrections, environmental changes, or new objectives.

A requirement engineering document is unambiguous if and only if every requirement has only one interpretation. Since, the developers and the customers are from different background and that's why they tends to understand things differently without knowing it . Natural language is the most used representation for stating requirements in industry and it is inherently ambiguous. The majority of requirement documents are written in natural language as the survey by Mich et al. shows [5].The customers and software developers may disagree on interpretations which leads to disastrous software failure. In software development the later an error is found, the more expensive its correction is. An empirical study by Kamsties et al.[6] shows that the ambiguities are misinterpreted more often than any other types of defects and if detected they need to be clarified as soon as possible.Ambiguity means that one phrase can be interpreted in several ways.Ambiguity in the code or in the specifications especially in the case of safe critical systems may have disastrous consequences . No less serious problems caused by ambiguous description of requirements which besides influencing the success of the project in technical terms may also have damaging consequences in contractual terms [7].The RE text consists of different categories of ambiguity that may vary from lexical ambiguity to semantic ambiguity and again semantic ambiguity to pragmatic ambiguity. The RE specific ambiguity is context dependent. And thus the background of software developer is very important and domain knowledge may also play a key role [8]. Moreover, lots of factor are responsible for introducing ambiguity in the RE text and thus complete and precise requirement specification is very difficult to achieve. Detecting ambiguity involves many sentences and requires semantic processing and understanding and is refer to pragmatic ambiguity which mainly context dependent and also depends upon the background of the stakeholders[9].

III. TYPES OF AMBIGUITY

Words and sentences in natural language corresponds to a vast number of meanings. Text information usually enables more than one interpretation . To find the one correct interpretation that should be programmed we need to interact with the customer and the domain expert. Each word may correspond to many different concepts and there may be more than one set of grammatical dependencies among the words in different sub phrases of the sentences.

The ambiguity in a text depends upon many factors and thus have different categories of ambiguity. Words and sentences in natural language may corresponds to a vast number of concept. Each word in a sentence may corresponds to many different meanings and there may be more than one type of grammatical dependencies among the words in different sub phrases of the sentence. It is basically ambiguity that provides flexibility and usability to any text and thus it cannot be eliminated. Also it is very important to mention the role played by the context which may influence the understanding of a phrase positively or negatively. And thus domain knowledge is a key factor that facilitates the understanding. But the presence of ambiguity makes the analysis more complicated since it requires greater cognitive effort. To obtain semantic information directly from a text , manual or automatic methods can be used.

a. *Lexical Ambiguity:* It occurs when a word has several meaning.
   Example : "GREEN"
   The word green has two meanings –
   " Green in colour " OR "Immature".

b. *Syntactic Ambiguity :* It occurs when a given sequence of words can be given more than one grammatical structure and each having a different meaning . In other words it means how similar are two words with respect to there syntactic function or role? It even helps to solve the problem of plagiarisms. The disadvantage of syntactic similarity is that two sentences having the same words in different order can have high syntactic similarity but a completely different meaning.
   Example : " SMALL CAR FACTORY "
   The above sentence can have two meanings .
   "(small car) factory " OR " (small) car factory ".

c. *Semantic Ambiguity :*It occurs when a sentence has more than one way of reading it within its context although it contains no lexical or structural ambiguity . To define semantics of a word the same words of two sentences have to be compared by including there context.
   Example : "ALL CITIZENS SHOULD HAVE A SOCIAL SECURUTY NUMBER ".
   The above sentence can be interpreted as :
   "Every citizen has an individual social security number ".
   " All citizens have same social security number ".

d. *Pragmatic ambiguity :* It occurs when a sentence has several meaning in the context in which it is uttered . It depends upon the background of the requirement engineers and thus have multiple interpretations. It influences the understanding of a phrase positively or negatively.
   Example: Do you want to have a cup a tea ?
   There is two meaning to the above question .
   Either an informative question — "Do you feel a desire to a cup of coffee?",
   Or a polite offer — "I can make you a cup of coffee if you want".

IV. APPROACHES TO MEASURE AND DETECT AMBIGUITY

a. Use of formal specification language (formal mathematical method) for stating requirement specification.

The formal methods are used to specify the requirements of any system. But still the initial requirements are always written in natural language and these informal requirements are translated in formal or semi – formal language by the requirement engineers.

The primary reason for using mathematical formalism in requirement engineering is the quality of the requirements. The ambiguity is the biggest problem within requirement engineering and the exact science can minimize or remove it [9].

The formal methods techniques can be divided into two main categories : *Model based methods and property based methods.*

The formal methods are based on the concept of set theory and logics. The first phase in the software development process is requirement elicitation and elaboration of the software requirement followed by analysis. This part of process of software creation is wrongly considered as easy one and not enough attention is given to it .

The formal methods are basically mathematics including logic and abstraction oriented specification which emphasizes properties rather than algorithms. And by design calculi we mean formal, mathematical in particular formal logical rules that apply to formal specification test[10,11].

The formal methods can also be applied at the different phases of software development and following goals can be achieved:

- Formal methods helps in writing specifications which are not always identical to the stated requirements . And this can be achieved by formal specification since it is unambiguous and we can also prove certain properties about it.
- It ensures that the implementation of a particular software as well as hardware product should satisfy the requirement specification.

- They are basically concerned for the development and maintenance of critical reliable system on time and within budget. It acts as evidence which ensures that the system is secure , reliable and correct .

b. Inspection technique (Based on checklist and scenario based reading).

The inspection technique is used to detect ambiguities in informal requirement documents before formal specification are produced in order to avoid interpretations, rather than searching for them afterwards.

The RE document is context dependent and can be observed only by a reader who has knowledge of the particular requirement context or of the other requirements. The inspection technique has got good potential to detect ambiguities.

Natural language is the most used representation for specifying requirements in the software industry. However, natural languages are ambiguous in nature. While inconsistencies and incompleteness can be mechanically detected in formal specification ambiguities in informal specification often results in incorrect specifications. If the inspection technique is used for detecting ambiguities in informal requirement documents before formal specification are produced and thus misinterpretations can be avoided [12].

Several inspection techniques have been proposed for spotting ambiguities.

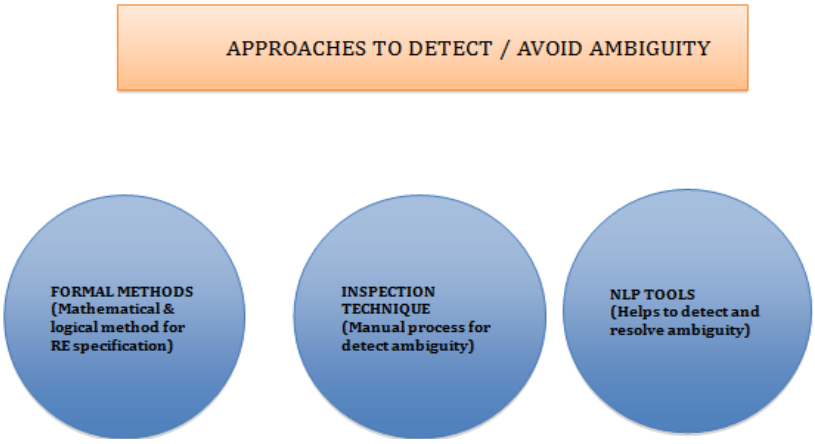
- The most effective approach is to hand over requirements to several different stakeholders, to ask each for an interpretation, and to compare these interpretations afterwards. If the interpretations differ, the requirements are ambiguous . This approach is economically feasible only for small sets of requirements.
- A detailed checklist of ambiguous words often used in requirements is provided in reference.These checklists help to find many linguistic ambiguities, but they do not address RE-specific ambiguities.
- Some other inspection techniques assume that inspectors are able to detect ambiguities just by reading; no guidance is provided on how to find an ambiguity. There is usually one checklist item asking, "is the requirement ambiguous?" The major problem of ambiguity is not being aware of it. Thus, simply asking whether there is an ambiguity is not much help.

The most recommended solution to the ambiguity problem is the use of a formal requirements specification languages or a semi-formal requirements specification languages, such as UML , rather than natural language. Such a language has a more-or-less well-defined semantics. Thus, the degree of ambiguity in requirements is at least significantly diminished if not eliminated. However, even when such a language is used, the initial requirements are written in natural language. These initial requirements must be translated into the formal or semi-formal language by a requirements engineer. During this translation, an ambiguous informal requirement is often not recognized as such, and it ends up becoming an unambiguously wrong formal or semi-formal requirement, if it is unconsciously misinterpreted. Such a misinterpretation can slip through undetected, because the client's domain experts are often not able to read the formal or semi-formal language well enough to detect a meaning different from their experiences or intentions. Simulation, another way of validating formal requirements, can show only the presence of misinterpretations but not their absence. The same criticism could be leveled against inspections. However, the relation between simulation and inspection for requirements validation or ambiguity detection can be compared to the relation between testing and inspection for detecting errors in code. Neither testing nor inspections can find all errors. Testing can be used to show the presence of errors but not their absence. Of course, it remains to compare the effectiveness of different methods to detect ambiguities.

c. Using Natural language tools (NLP tools) for measuring and detecting ambiguity.

Language is the primary means of communication used by humans. Natural language processing (NLP) is concerned with the development of computational model of aspects of human language processing . There are number of factors making NLP difficult and they relate to the problem of representation and interpretations. Machine translation is the first application area of NLP. It involves complete linguistic analysis of an RE text which is written in natural language and linguistic generation of an output sequence .

Natural language processing (NLP), also known as computational linguistics, is a set of models and techniques for analyzing text computationally. The achievement of the quality of software requirement is the first step towards the software quality. The process that leads to the quality of requirement starts from the analysis of the requirements expressed in natural language and continues with there formalization and verification. Hence it is very important to provide methods and tools for the analysis of the RE text. Our main aim or objective is to device an automatic tool that detects ambiguities, inconsistencies etc from RE text.Ambiguity is an intrinsic phenomenon of natural language which means the capability to understand any sentence or words in multiple possible ways. Identification of ambiguous words and phrases is a crucial aspects since 71.8% of RE text are written in natural language. Many NLP tools are available which are trying to resolve the problem of ambiguity to some extent. But since the ambiguity depends on many factors thus complete removal is not possible, but at least we can minimize the ambiguity to some extent.The methods followed by NLP tools are to identify the sentences in any text which are prone to ambiguity. And for each ambiguities make the user understand the reason for ambiguity so that the user can resolve ambiguity.And the final decision depends upon the user . But still the users background and his/her domain knowledge also influence the decision , which is called as *pragmatic ambiguity*. Since detecting ambiguity in some cases involves references, processing and understanding of many sentences and thus concept of pragmatic ambiguity arises [13].



V. RELATED WORK/LITERATURE SURVEY

Several studies dealing with the ambiguity measure and detection is available in the literature. We will briefly discuss some of those we consider to be particular interest.

Manual detection is the most popular approach to detect, resolve and measure ambiguities in the RE text. In one of the early studies Meyer [4] suggested those areas in the natural language were the chances of doing mistakes is obvious. And he gave stress to the fact that the RE text is always written in natural language and by using proper formal specification method is absolutely necessary.Another study worth mentioning here is done by Kamsties et al [6] who introduces the five different categories of ambiguity and they are – Lexical ambiguity, Systematic ambiguity, Discourse ambiguity, Referential and Domain ambiguity. By explaining the steps to detect ambiguity they mainly argued in the favor of manual inspection [12].It offers to a requirements engineer an efficient inspection technique for detecting ambiguous requirements that is applicable in industrial RE. Second, it offers an approach to identify ambiguity types that can occur in a particular RE context. Realistically, one cannot expect to identify types of ambiguity that he or she never ever has thought about or come across. Rather, the contribution lies in the systematic way to explore this implicitly existing knowledge by using the heuristics and in increasing the requirements engineer's awareness of the problem.

Many other studies tries to reduce the problems associated with unrestricted natural language by limiting the scope of the language.

Macius and Pulman [14] mainly applies context independent techniques to control the production of RE text. They uses the techniques to control the vocabulary and style of writing used and it should follow certain set of rules for writing any RE text to make it simple and clear to understand.Goldin and Berry [15] implemented a tool which extracts abstraction from RE text of repeated segments identifying significant concepts on the application field of the problem at hand . But is mainly restricted to lexical analysis only.

I Hooks [16]discusses a set of quality characteristics necessary to produce well defined natural language requirements. The paper mainly highlights the sources of defects in RE text and to identify the related risks. The paper presents some common problems that arises when requirements are produced and looks at how to avoid them . It provides as in depth survey of the principal sources of the defects in NL requirements and the related risks .Wilson et al [17] examines thequality evaluation of natural language. It defines a quality model composed of quality indicators and attributes to develop an automatic tool to perform the analysis against the quality model aiming to detect defects and collect metrics. But there work never discusses the important issues like adaptability of the tools for different categories of domains. They only considers the syntactical and structural aspects of the requirement and never considers the semantic details of the RE document .

Several studies give much effort on handling ambiguity in RE text [18]. Kamesties and Paech [19, 12] mainly gives emphasis on the ambiguity evaluation of RE text. They provided a checklist addressing not only linguistic ambiguity but also ambiguity related to particular domain.They begin from particular domain of the application under development . They mainly identified the principal deficiencies of the solutions to avoid NL requirements .Mich and Garigliano [5] gives set of measures for sementic and syntactic ambiguity in RE text. Their approach is based on the use of information on the possible meanings and roles of the words within a sentence and possible interpretation of sentence.Natt Och Dag et al [20] gives an approach based on statistical techniques for the similarity analysis of NL requirements aimed at identifying duplicate requirement pairs.

Several studies are also considering the concept of pragmatic ambiguity. A Ferrari et al [8,21]discusses about the pragmatic ambiguity in RE text which are context dependent. They presented a approach where a graph based modeling of background knowledge of different readers are generated and then uses a shortest path search only to model pragmatic implementation.

Ishraar Hussain et al discusses about the concept of text classification system to detect ambiguity in an SRS document by classifying the passage as ambiguous or unambiguous [22].Tyler Baldwin et al [23] introduces the term ambiguity term detection task which detects whether term is ambiguous relative to a topical domain. Unlike other concepts it makes general ambiguity judgments about terms, rather than resolving individual instances.Ayan Nigam , Neeraj arya et al [24] designed a tool that detects lexical , syntax and syntactic ambiguity and determines ambiguity in percentage basis that helps analyst to identify which ambiguity is present in highest percentage.

Fabbrini F & Fusani M et al [25] presented a method and a tool for evaluating natural language software requirements according to previously designed quality model . It supports a very critical phase of the software process the passage from informal requirement (written in NL) to semi-formal/formal models.Benedikt , Olive et al [26] explained that it is not only important to detect ambiguity but it is also important to educate the analyst by explaining them about the source of ambiguities . The ambiguity detection tool by author detects four different types of ambiguity. This tool is helpful since it educates the analyst. It has good potential to save cost and time thus improves the RE text.Text classification is also a good technique for identifying and resolving ambiguity. Automatic text classification has been considered as a important method to handle and process a vast amount of document in digital forms. Today text classification is a necessity due to the very large amount of text documents that we have to deal with daily . M. Ikonomakis, S. Kotsiantis et al discusses that for a specified classification method, classification performances of the classifiers based on training text corpuses. It is a problem of Artificial Intelligence research topic, especially given the vast number of documents available in the form of web pages and RE text [27].Text classification is one of the key techniques in text mining to categorized document in supervised manner. Meneka S et al discusses that the processing of text classification involves two main problems are the extraction of feature terms that become effective keywords in the training phase and then the actual classification of the document using these feature terms in the test phase . The proposed method use text mining algorithms to extract keywords from journal papers. Then the documents are classified using machine learning algorithms - Naïve Bayes , Decision Tree and k – Nearest neighbor [28].

VI. FUTURE RESESRCH DIRECTION

One of the most important phases of SDLC is requirement gathering and the rest of the project depends upon this initial stages of software development. And if the requirements are not specified properly then the RE text is ambiguous and thus the whole product will get affected. And one method to resolve this problem is to detect and resolve ambiguity at early stages of development. Till now all the available tools only detects the lexical, syntactic and syntax ambiguity.By formal methods which is a mathematical technique for specifyingrequirements in any document is very important for development of any hardware and software. There is not much published evidence to support one side or the other, and a lot of what is said about formal methods is based on assertions and notfacts. Thus, some of the beliefs about formal methods have been exaggerated and have acquired almost the status of myth [29,30]. It is quite simple and clear that we can remove the lexical and syntactic ambiguity. In future work other types of ambiguities such as semantic and pragmatic ambiguities can also be considered which depends upon the conceptual interpretation of any statement and thus requires domain knowledge . And since it is context dependent depends upon the background of the stakeholders [31,32].A pragmatic ambiguity in a requirement if different readers give different interpretations to it depending upon the context of the requirement. The context of the requirement includes the other requirements of same document which influence the understanding of the requirement [33,34].The approach is based on the concept of constructing Domain Knowledge Graph . For the given set of documents in the domain of the requirements, a knowledge graph is a weighted graph that includes the concept of domain documents , and the relationship among concepts [34,35].

The inspection technique has good potential to detect and resolve ambiguity in industrial requirement engineering[36]. It offers an approach to identify ambiguity types that can occur in particular RE text. The future work aims at investigating in how much meetings increase the number of detected ambiguities . We must also determine which meeting formats allow reviewers to best exchange there interpretations of requirements [37].

It is well known that presence of inaccuracies in requirement documents and specification introduces serious risks to all the consequent phases of software development . And thus it is important to provide tools and method for the analysis of natural language requirement document. Several studies dealing with the evaluation and achievement of the of quality in natural language specification can be found in literature [38].

There are some tools in requirements management that transforms unstructured texts of requirements into structured texts.

They neither build any model nor generate refining questions like any other tools [39].Even if it is widely believed that the graphical representation is the best method for the communication between the analyst and the user there are some serious doubts about it . It has been found that even a co-operation between experienced analyst and sophisticated users who are not familiar with the particular graphical language results in semantic error rates of about 25% for entities and 70% for relations [40].There are number of research methods for requirement elicitation and analysis of requirement based on existing domain ontologies.For example [41] uses a domain ontology and requirements meta model to elicit and define textual requirements. The system GOORE proposed in [42] represents a approach to goal oriented and ontology – driven requirements specification. It represents the knowledge of a specific domain as an ontology for goal oriented requirement analysis. A major disadvantage is the need for pre existing ontology and there is no suitable method for building this ontology for requirement elicitation.The RE text in many situations have multiple interpretations and thus Gorfein S suggested a scheme to define indices of ambiguity both structural and semantic. It also investigates the feasibility of calculating these indices using the knowledge base of natural language processing system and the outputs produces during the text parsing phase [43]. The SRS document are the medium used to communicate with the customers for developing any software . The process sometimes takes considerable amount of time to specify documents that can range from words , phrases and sentences where each can be potentially interpreted wrongly . The author Gnesi S, Lami G et al et al suggested a quality model for the RE text and text classification system to automate the quality assessment process and they uses the NLP techniques to asses the quality of RE text. The NLP text classification technique was employed to build a system for automatic detection of ambiguity in RE text in terms of quality indicators defined in the

quality model. Thus the author uses a decision tree based classifier , trained with and tested against human annotated samples . This is done by considering machine learning algorithms [44,45]

VII. FINDINGS/CONCLUSION

Lexical and syntactic ambiguity is easy to remove.This can be done by flagging text as "Ambiguous" or "Unambiguous". But semantic and pragmatic ambiguity is very difficult to remove since itdepends upon the background of the stakeholders. And even the conceptual interpretation of any statement depends upon the background of the stakeholders and thus pragmatic ambiguity is also a very important factor.The RE text are context dependent. And thus by improving the awareness about a particular domain we can also detect ambiguity.The ambiguity in RE text can be reduced by the process of inspection. It has very good potential to detect ambiguity.By following the proper formal specification method for writing any RE text also helps to improve the quality of RE text.By lowering the level of ambiguity in the RE text will lead to a better quality conceptual description of the solution and also reduces the amount of time required for requirement analysis and specification.There are many approaches to detect/avoid ambiguity. And the degree of ambiguity in requirement cannot be completely removed from the RE document but at least can be significantly diminished.Thus we can work in three different areas in order to detect, measure, and avoid ambiguity.

- Properly design or specify the requirement engineering specifications by using formal methods which are mathematical and logical method for writing or specifying RE text.
- To resolve ambiguity problem we can use formal requirements specification languages or a semi-formal requirements specification languages, such as UML, rather than natural language.
- By following the inspection techniques which is a manual method to detect ambiguity can also be used for specifying RE text .
- By using NLP tools to detect / measure and avoid ambiguity. The tool mainly detects syntactic and lexical ambiguity . But semantic and pragmatic ambiguity is also very important since they are context dependent and the background of the requirement engineering and stakeholders also plays a vital role in specifying RE text.
- Our future work aims at investigating in how much meetings increase the number of detected ambiguities. In meetings, perhaps ambiguities that slipped through individual preparation can be detected. We must determine which meeting formats allow reviewers to best exchange their interpretations of requirements.

REFERENCES

[1] Rajib Mall . " Fundamentals of Software Engineering ", 2009.

[2] R S Pressman & Associates Inc. " Software Engineering – A Practitioner approach 2001.

[3] Boehm , B . "Tutorial Software Risk Management (1989) , IEEE Computer society press.

[4] Meyer B," On formalism in specifications", IEEE software,2(1) , January 1985 , 6 – 20.

[5] L Mich and R Garigaliano , " Ambiguity measure in requirement engineering ",in Proc of ICS '00 , 16th IFIP WCC 2000 , pp 39 -48, August 2000.

[6] Kamsties,E.,Knethen,A.V.,Philipps,J.,Schaˈtz,B "An empirical investigation of the defect detection capabilities of requirements specification languages. "In: Proceedings of the Sixth International Workshop on Evaluation of Modelling Methods in Systems Analysis and Design (EMMSAD 2001), pp. 125–136 (2001) .

[7] O S Borne M , Macnish C K ,"Processing natural language software requirement specification ", Proc , of the International conference on requirement engineering 1996 (ICRE '96), pp . 229 – 236.

[8] Alessio Ferrari , Giuseppe Lipari , Stefania Gnesi , Giorgio O Spanglo ," Pragmatic Ambiguity Detection in Natural Language Requirements ", In 1st International workshop for Artificial Intelligence for Requirement Engineering ,pp -1-8 , IEEE(2014).

[9] Ian Sommerville . Software Engineering , 6th Edition. Addison –Wesley 2002.

[10] Galambos , Abelson , Black , "Knowledge Structure " , LEA 1986.

[11] Boehm B W : Software Engineering Economics : Prentice Hall 1981.

[12] E Kamsties , D M Berry and B Paech " Detecting Ambiguity in requirement document using inspection " In : Workshops on inspection in software engineering ,Paris France (2001) 68 – 80.

[13] K . Ryan , " The role of natural language in requirement engineering in natural language ," in Proc. Of ISRE'93, 1993,pp.240-242.

[14] B . Macias , S G. Pulman . Natural language processing for requirement specification . In Redmill and Anderson , Safety Critical Systems, pages 57 – 89 . Champman and Hall 1993.

[15] L Goldin and , DM Berry . Abstfinder , a prototype Abstraction finder for natural language text for use in requirement elicitation : Design, Methodology and evaluation. First international conference in requirement engineering , 1994.

[16] I Hooks , Writing good requirements , Proc. Of the fourth International symposium of the NCOSE , 12994, Vol. 2, pp197-203

[17] W M Wilson , L H Rosemberg , L E Hyatt . Automated Quality Analysis of Natural Language Requirement Specification . PNSQC Conference October 1996.

[18] N.E Fuchs , R . Schwitter Specifying Logic Programs in Controlled Natural Language, Workshop on Computational Logic for Natural Language Processing , Edinburg, April 3-5 , 1995.

[19] E. Kamsties , B Peach Taming Ambiguity in Natural Language Requirements ICSSEA 2000 – 5.

[20] J Nattoch Dag , B Regnell P. Carlshamre , M Andersson, J Karlsson . Evaluating Automated Support for requirements Similarity Analysis in Market – Driven development Seventh International workshop on Requirement engineering foundation for quality , Interlaken Switzerland , June 4-5 2001.

[21] A Ferrari and S Gneisi , " Using collective intelligence to detect pragmatic ambiguities ", in Proc of RE'12,2012,pp191-200.

[22] I Hussain , (2007) , " Using text classification system to Automate Ambiguity Detection in SRS document ". Masters Thesis , Computer science and software engineering Department , Concordia university , Montreal Canada , August.

[23] Tyler Baldwin , Yunyao Li, Bogdan Alexe and Ioana R Stanoi , " Utomatic Term Ambiguity Detection", In Proc of the 51st Annual Meeting of the Association for Computational Linguistic, page 804 – 809 , Bulgaria , August 2013.

[24] Ayan Nigam , Neeraj Arys , Bhawana Nigam and Deepika Jain , " Tool for Automatic Discovery of Ambiguity in Requirements ", International Jouranal of Computer Science Issues , Vol 9 , Issue 5 , No. 2 , September 2012.

[25] Fabbrini F , Fusani M , Gnesi S , Lami G (2001) , " The Linguistic approach to the Natural Language Requirements , quality : benefits for the use of automatic tool. In:Proceedings of the twenty sixth annual IEEE computer society - NASA GFSC software engineering workshop , pp 97-105.

[26] Mona Batra , Amit Malik , Dr . Meenu Dave, Formal methods: Benefits, Challenges and future direction ", Journal of global research in computer science , volume 4 , No. 5 , May 2013.

[27] 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.

[28] S Meneka , Radha N , " Text Classification using Keyword Extraction Technique ",International Journal in Advance Research in Computer Science and Software Engineering , Volume 3 , No. 12 , December 2013.

[29] F. Lehner , "Quality control in software documentation based on processing and management" , vol;29, No. 5 , pp 551 – 568 , 1993.

[30] Kiyavitskaya,N.,Zeni,N.,Mich,L.,Berry,D.M.:Requirements for tools for ambiguity identification and measurement in natural language requirements specifications. Requir. Eng. 13, 207–239 (2008)

[31] Boehm B , W : Software engineering Economics . Prentice Hall, 1981.

[32] D . M . Berry , E Kamsties , " The syntactically all the plurals in specifications ,"IEEE software , vol 22, no. 1 , pp. 55-57, 2005.

[33] L Mich , M Franch and P N Inverardi , " Market research for requirement analysis using linguistic tools ," Requir. Eng. Vol 9, no 1.pp 40 -56 , 2004.

[34] C Faloutos , K S McCurley , and A Tomkins , " Fast discovery of connection subgraph ," in Proc. Of KDD '04,pp. 118-127.

[35] P. N Tan, M Steinbach , and V kumar , Introduction to Data Mining. Addison – Wesley , 2005

[36] Sommerville . I and Sawyer. P , Requirement Engineering : A good Practice Guide , Jhon Wiley & Sons (1997)

[37] Allen . J , Natural Language Understanding , Addison – Wesley , Readin g , MA (1995) Second Edition.

[38] J . Krogstie , O . I Lindland, G Sindre . "Towards a deeper understanding of quality in Requirement Engineering ". In 7th International CAISE Conference , vol.932 of Lecture Notes in Computer Science , pages 82-95,1995.

[39] Coad, P., Yourdon, E.: Object-Oriented Analysis. Prentice Hall, 1991.

[40] Petre, M.: Why looking isn't always seeing: Readership skills and graphical programming. Communication of the ACM, Vol. 38, No. 6, pp. 33-42, 1995.

[41] Lee,Y.,Zhao,W. "An Ontology Based Approach for Domain Requirements Elicitation and Analysis." In: Proceedings of the First International Multi-Symposiums on Computer and Computational Sciences, 2006

[42] Shibaoka, M., Kaiya, H., Saeki, M.: GOORE: Goal-Oriented and Ontology Driven Requirements Elicitation Method. In: Hainaut, J.-L., Rundensteiner, E.A., Kirchberg, M., Bertolotto, M., Brochhausen, M., Chen, Y.-P.P., Cherfi, S.S.-S., Do- err, M., Han, H., Hartmann, S., Parsons, J., Poels, G., Rolland, C., Trujillo, J., Yu, E., Zimanyie, E. (eds.): ER Workshops, LNCS, Vol. 4802, pp. 225234, Springer, 2007.

[43] Gorfein S . (ed) , Resolving Semantic Ambiguity , Springer – Verlag , 1989

[44] Gnesi S , Lami G , Trentani G, Fabbrini F & Fusani M ," An automatic Tool for the Analysis of Natural Language Requirements". International Journal of Computer System Science and Engineering , Special issue on Automated Tool for Requirement Engineering , 20(1), Leicester: CRL Publishing.

[45] Quinlan , J .R (1993). C4.5: Programs for Machine Learning . San Mateo, CA:Morgan Kaufmann.

