IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 6, June 2016

Identification of Ambiguity in Requirement Specification using Multilingual Word Sense

T. Jaison Vimalraj[1], **B. Seema**[2]

Teaching Fellow, Department of Information Technology, University College of Engineering, Trichy, India [1]

Student, Department of Software Engineering, University College of Engineering, Trichy, India [2]

Abstract: Requirements are the basic building block for development of good product. Requirement analysis is one of the major steps to reduce complexity of developing product. Several automatic analysis methods have been proposed to improve the quality of requirement. Currently there is no technique available to reduce ambiguity caused by Lexical, Syntactic and Syntax ambiguities. In this paper we identify these ambiguities and reduce them to improve the quality of requirement.

Keywords: Requirement Specification, Lexical Ambiguity, Syntactic Ambiguity, Semantic Ambiguity.

I. INTRODUCTION

Requirement gathering is one of the important steps in software development. Requirements should be clear in order to understand by the development team. Development team will not spend more time in analysing the requirement. If the requirement is not clear then the development team will create improper outcome. Ambiguity is the main factor which affects the quality of the product.

Ambiguity in requirement specification will affect the development of product. If ambiguity present in requirement specification then that will be considered as a bug. This will lead to poor design and performance of the final product. If ambiguities present in requirements where identified in requirement analysis itself there is no need to spend lot of time and money to rectify in development phase.

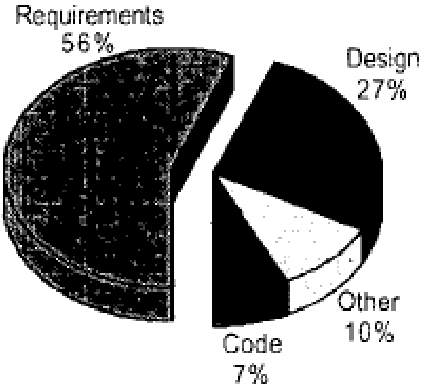

Fig 1: Distribution of bugs in different phases of Development cycle [1]

Testing the requirement is one of the important steps in software engineering. Requirement testing means verification and validation of software requirements [2]. It is important to identify and resolve software problems in software life cycle [3].

The severity of the ambiguity problem is emphasized by several authors [4], [5]. [6] Proposed alpha-beta procedure to cut off the branches of requirement tree and reduce the complexity of tree traversal.

In the rest of the paper, section II describes the background of our project which includes related work for requirement analysis. Section III describes the system model for requirement analysis. Section IV describes the conclusion and future enhancement for this paper.

II. BACKGROUND

This section explains related requirement analysis techniques. The problem of synonyms and homonyms is occurring in various areas of system analysis and design including requirement engineering [4], [5]. [7] Utilize natural language pattern for ambiguity detection. Metrics can also used to identify and rank ambiguities [8]. [9], [10] employ machine learning approach to find ambiguity in requirements. Several methods can use mental capabilities of humans to resolve these ambiguities [11], [12].

Companies need to maintain large number of process models [13], here human not able to resolve ambiguities. [14] Describes five major factors of requirement failure as follows

- Failure to effectively manage conflict.
- Lack of clear statement about the design to be solved
- Too much unrecognized disambiguation
- Not knowing who is responsible for what
- Lack of awareness of requirements risk

To improve the quality of given process in information system several authors utilize the technique of integrating process modeling with requirement engineering [15], [16]. Natural language pattern can also be utilized for ambiguity detection [12], [17].

DOI 10.17148/IJARCCE.2016.5681

III.PROPOSED APPROACH

This section describes the overall requirement analysis technique to reduce ambiguity.

A. Overview

The analysis of requirement involve following four main components

- Requirement Specification.
- Detecting Ambiguous Words
- Parts of Speech Tagging
- Multilingual Word Sense

By employing the above components we can identify three types of ambiguity like lexical, syntactic and semantic. Once these ambiguities detected we can able to reduce ambiguity in overall system specification.
The following system represents the overall methodology of ambiguity detection in requirement specification

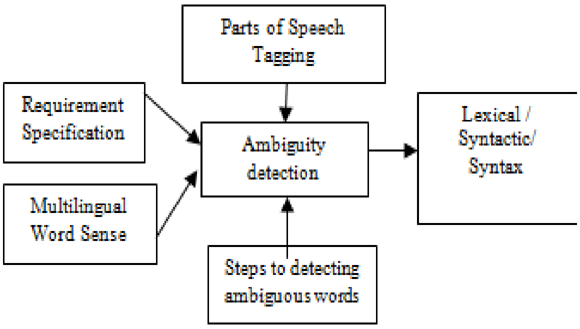
Parts of Speech Tagging
Requirement Specification
Ambiguity detection
Lexical / Syntactic/ Syntax
Multilingual Word Sense
Steps to detecting ambiguous words

Fig 2: overall system of ambiguity detection

Requirement specifications are the major building block of software development so we analysing it using ambiguity detection methodology. For that purpose we are using multilingual word sense approach to identify sense of words (In which sense the given word used). It started with the first step as POS tagging which is used to portioning the sentence into words. Ambiguity detection involves the process of classifying ambiguity as Lexical, Syntactic and Syntax. This methodology not removing ambiguity completely but reducing the ambiguity.

B. Requirement Specification

The goal of requirement specification is to describe what system to build. Requirement specifications play many roles as follows

- Requirement Specification is the bridge between the customer and developer which defines what to build.
- It defines constraints to design and development.
- It is the basis for estimating cost and schedule.

C. Multilingual Word Sense

It is the main component of ambiguity detection. Ambiguous words are identified by using word sense approach. It will evaluate different types of ambiguous words by referring the dictionary. For example we consider the word application which is having following different senses

Sense-1 Application: A program which is run in computer to perform some task.
Sense-2 Application: Medical task when antiseptic applied to skin.
Sense-3 Application: Written request for employment or any admission.
The above word application having more than one meaning which is considered as synonyms and it may also lead to ambiguity.

D. Parts of Speech Tagging

Parts of speech tagging is the important step in ambiguity detection. POS tagging involve the process of dividing the sentence in to following categories.

- Noun
- Adverb
- Adjective
- Verb
- Preposition

E. Ambiguities in Requirement Specification

Ambiguities should be identified in requirement specifications. Generally ambiguities divided in to three categories as follows

Lexical Ambiguity-It is basically defined as a word having more than one meaning. For example we consider the word “black” which means “dark” or it may be “corrupted”. It may also occur when two words having same pronunciation. For example consider word “too” and “two” having the same sound of pronunciation but meaning is different.

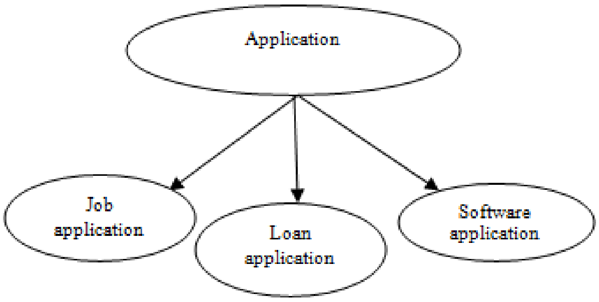

Fig 3: Hyponymy Tree

Syntactic Ambiguity-Syntactic ambiguity also called as structural ambiguity. It means that the given sequence of word make different grammatical structure and each one giving different meaning. For example “Small toy shop” which means in both senses as (small toy) shop and small (toy shop).

Syntax Ambiguity-It occurs due to the following two main factors

- If any sentence not end with (.).
- If user element not specified.

F. Steps for Detecting Ambiguous words

It is used to classify ambiguity based on the factors like syntax, syntactic and lexical ambiguity.

DOI 10.17148/IJARCCE.2016.5681

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 6, June 2016

1. Store the multilingual word sense dictionary in the database.
2. Analyze the requirement specification to find the ambiguity.
3. Each line of requirements checked with the multilingual word sense backend to find ambiguous words.
4. The ambiguous words are then compared with POS tagging to classify ambiguous words as lexical, syntactic and semantic ambiguity.
5. After identifying ambiguities we have to reduce ambiguity by replacing those ambiguous words by alternative unambiguous words.

G. Ambiguity Detection

Ambiguity detection involve in the process of analyzing the requirement. First we have to give the requirements as input. After that requirements processed by POS tagging to find ambiguous words like syntax, lexical and syntactic. The ambiguous words are analyzed by referring the multilingual word sense. Those ambiguous words are than reduced by replacing those words by less ambiguous words.

IV. CONCLUSION

The paper presents the analysis of requirements using ambiguity detection methodology. This can be evaluated using multilingual word sense, pos tagging. It involve in detecting lexical, syntactic and syntax ambiguity. In future semantic Ambiguity, vagueness ambiguity, incompleteness, referential, Domain ambiguity and pragmatic ambiguity can also be considered.

Semantic Ambiguity- It occurs only when the sentence having more than one interpretation. For example "Mouse and keyboard are parts of computer". Here another context is needed to know whether Mouse and Keyboard are belong to same computer or not.

Vagueness Ambiguity- It occurs only when the given sentence is not giving clear meaning. For example "Software is a platform for project development". Here platform can define more than one meaning.

Incompleteness- It occurs only when the given sentence not giving complete meaning. For example "Syntax error". Here Syntax error not defining which program having syntax error.

Referential- It occurs only when the grammatically correct sentence with reference which confuses the reader.

Pragmatic ambiguity- It occurs when the whole sentence having more than one meaning.

Domain ambiguity- It occurs when the application details are not clear.

Based on the above definition criteria conditions should be evaluated for identifying semantic, vagueness, incompleteness, referential, domain and pragmatic ambiguity. Checklist can also create to reduce complication while identifying ambiguity.

REFERENCES

[1] Gery mogyorodi, starbase corporation, "requirement based testing: an overview". 2001 IEEE

[2] Yonghua Li, Fengdi shu, Guoqing Wu, Zhengping liang "A requirement engineering for embedded real time software-SREE," wuhan university journal of natural science, Vol. 11, No.3 , 2006, PP. 533-538.

[3] Bary W. Boehm, TRW, "Verifying and Validating Software requirements and design specification" , January 1984 IEEE.

[4] E.Kamsties, "Understanding ambiguity in requiremens engineering," in Engineering and Managing Software Requirements, A. Aurum and C.Wohlin Eds.New York, NY, USA: Springer, 2005, pp.245-266.

[5] D.M.Berry, E. Kamsties, and M.M. Krieger. (2003). From contract drafting to software specification: Linguistic sources of ambiguity A handbook [online]. Available: http://se.uwaterloo.ca/dberry/handbook/ambiguityHandbook.pdf.

[6] Gang Liu, Shaobin Huang, Xiufeng Piao, "study on requirement testing method based on alpha beta cutoff procedure" college of computer science and technology, Harbin Engineering University, Harbin, Heilongjiang, China, 2008 IEEE.

[7] B. Gleich, o. Creighton, and L. Kof, "Ambiguity detection: Towards a tool explaining ambiguity sources," in requirements Engineering: Foundation for Software Quality. New York, NY, USA: Springer, 2010, pp.218-232.

[8] Y. Wang, I.L.M Gutierrez, K. Winbladh, and H. Fang, "Äutomatic detection of ambiguous terminology for software requirements," in Natural Language Processing Information Systems. NewYork,NY, USA: Springer, 2013, pp.25-37

[9] H. Yang, A.De Roeck, and B. Nuseibeh, " Automatic detection of nocuous coordination in natural language requirements," in Proc. IEEE/ ACM Int.Conf. Autom. Softw. Eng., 2010, pp. 53-62.

[10] H. Yang, A. De Roeck, V. Gervasi, A.willis, and B. Nuseibeh, "Analysing anaphoric ambiguity in natural language requirements," Requirements Eng., vol. 16, no.3, pp.163-189,2011.

[11] F. chantree, B. Nuseibeh, A. De Roeck, and A. willis, "Identifying nocuous ambiguities in natural language requirements," in Proc. IEEE 14th Int. Conf. Requirements Eng., 2006, pp.59-68.

[12] B. Gleich, O. Creighton, and L. Kof, "Ambiguity detection towards a tool explaining ambiguity sources ," in requirements engineering: Foundation for Software quality. New York, NY, USA: Springer, 2010, pp. 218-232.

[13] M. Rosemann, "Potential pitfalls of process modeling: Part A," Bus. Process Manage. J., vol. 12, no. 2, pp.249-254, 2006.

[14] Gause, D.C., "User DRIVEN Design—The Luxury that has Become a Necessity, A Workshop in Full Lifecycle Requirements Management", ICRE 2000 Tutorial T7, Schaumberg, IL (23 June 2000).

[15] J. Barjis, "The importance of business process modeling in software systems design," Sci., Technol. Eng., 2003, PP.73-87, 2008.

[16] M. Attaran, "Exploring the relationship between information technology and business process reengineering," Inf. Manage., vol. 41, no.5, pp.585-596, 2004.

[17] C. Denger, D.M.Berry, and E.Kamsties, "Higher quality requirements specification through natural language patterns," in Proc.IEEE Int. Softw.: Sci. Comput. Program, vol. 71, no.1, pp.73-87, 2008.

DOI 10.17148/IJARCCE.2016.5681