Software requirement specifications documentation is one of the critical factors of the success of a software project. Imprecise, ambiguous or incomplete requirements lead to a product which is not acceptable by the users.

To express the views or thoughts, the universal method is the use of Natural Languages (NL). One of the intrinsic properties of Natural languages is ambiguity; hence, there are chances that the information represented by using Natural Languages like Hindi, English etc may be ambiguous. The term ambiguity indicates more than one different understanding of a sentence interpreted by different readers.

Ambiguity is a bottleneck of a variety of Natural Language Processing (NLP) applications like sentiment analysis, machine translation, opinion mining, information retrieval systems, search engines etc. Ambiguity is more dangerous in case of formal and technical documentation such as software requirements specification (SRS) documents, as misapprehended or imperfect requirement specifications are among the key causes of failure of any software project. For the success of a software project, it is very important that the SRS document exactly states what the stakeholder wants. The present thesis work focuses on development of an approach towards ambiguity resolution in software requirements specification documents.

This chapter deals with the overall structure of the thesis work. This chapter helps the readers to understand the exact problem statement, motivation and objective behind taking up this thesis, and the scope of this thesis.
1.1 Introduction

Requirement elicitation and documentation is one of the decisive phases of a software development life cycle. Studies have found that a critical root cause of a software project failure is the poor quality of the SRS document. Roughly 71.80% of software requirements are usually expressed in natural languages like English (Mich Luisa, 2004), they can be ambiguous and inconsistent as ambiguity is a fundamental characteristic of natural languages. Hence result of the requirements, represented in any natural language, of an SRS document is inconsistent and incongruous software designs and implementations.

Ambiguous requirements are often seen as a cause for project failure. Ambiguity is one of the most important problems in requirements specification documentation. In this work, we will disregard the deliberate ambiguity or the ambiguity that present in the early stages of requirements elicitation and will focus on ambiguity that remains in a so-called final natural language requirements document. Ambiguity is a key issue in the development of a software product because the different readers of the requirements specification may interpret the requirements differently. If the implementors' understanding of the document differs from that of the stakeholders or the end users, then they are likely not to be satisfied with the implementation produced by the implementers.

Many times, ambiguity is not noticed by anyone looking at the requirements document. Time and again, each reader subconsciously disambiguates the requirement specifications document to the interpretation which first comes into his or her mind and he or she thinks that this first understanding is the only interpretation of the document, without thinking of other possible interpretations.
A lot of the work has been done by the researchers to rendering the significance of SRS documents for success or failure of a software project. Researchers have acknowledged different areas of an SRS document, which are accountable for the success or failure of a software project. Gause D C, (2000), provides 5 most vital factors that are accountable for failure of efficient implementation of requirements. SRS is one of these critical factors. (Faulk, 2013), discussed the roles of SRS in large system and how much important it is in coordinating different stakeholders for building the right system. Mayer (1985), presented a systematic explanation of vital areas of an SRS, where SRS author is more likely to make mistakes. He categorized these areas into seven distinct classes named as “seven sins”. All these 7 areas are critical and affect the quality of an SRS document and Ambiguity is one of these 7 sins reported by him.

This thesis work focuses on development of an approach for efficient disambiguation of the SRS documents. The work describes the issues pertaining to the ambiguities in a document and tries to develop a technique for resolving Lexical ambiguities and ambiguities due to vagueness.

1.2 Problem Description

Software requirement specifications documentation is a critical phase of software development life cycle in the success or failure of a software project. The key medium for agreement between all the stakeholders of a software project to make a decision precisely what system is to be developed is An SRS document. An SRS document specifies what features/properties the developed system must have and how it functions. SRS document also provide control on system design and implementation. SRS document also helps as a testing
tool for validating the performance of the developed system. According to Standish Group’s CHAOS report (1995), a study of 175,000 projects, more than 50% projects were "challenged," with reduced functionality, cost-overrun and beyond the expected schedule. The top three project challenged factors are:

- Lack of user input (12.8%).
- Incomplete requirements and specifications (12.3%).
- Changing Requirements and specifications (11.8%).

The report, called Strategies for Project Recovery (Strategies for Project Recovery - PM Solutions, 2011), covers 163 companies, indicates that almost 37% of the software projects are at risk every year. It also shows 5 top causes of troubled projects and Requirements (Unclear, contradictory, ambiguous, imprecise) is one of them. So to deliver a quality software project, unambiguous SRS document is an important input.

Word sense disambiguation (WSD) is the ability to working out which sense of a Polysemy word (having multiple meanings) is triggered by its use in a particular context. WSD is usually performed on bags of words, (i.e., collections of naturally occurring words). If we disregard the punctuation, we can observe a text T as a sequence of words (w1, w2, . . . , wn), and WSD can be formally articulated as the task of allocating the most appropriate sense to all or some of the words in T according to the context in which they are used. Vagueness occurs when the user is not able to express himself clearly. Vagueness is inevitable in Natural Language. Very often people give specifications like “normal temperature”, “its cold outside” etc. these types of statements can be interpreted differently by different readers as per their understanding of the system.
The proposed work focuses on development of an approach for efficient disambiguation of the SRS documents.

The aim of this work is:

i) To study the vital role an ambiguous SRS document plays in the development of a software project.

ii) To study different types of ambiguities.

iii) Development of an approach which can effectively minimize the chances of ambiguities in a SRS document.

1.3 Motivation

The motivation behind taking up this domain as my research is that, the SRS documentation is the most effective phase of the software development life cycle (SDLC) and one can deliver quality software by focusing on the requirement engineering.

Resolving the present ambiguities in requirement engineering is very important for the following reasons:

i) Ambiguity is one of the major causes of software project failure.

ii) Ambiguity is not acceptable at all, because misinterpretations can lead to disastrous consequences in software design & implementation.

iii) Ambiguity causes scope creep, missed deadlines, run-away costs, under/over utilization of resources etc.

iv) The low rate of success of software projects is mainly due to the lack of clear and precise requirements.
v) The ambiguous or imprecise requirements may cause problems at any stage of the development of software project.

vi) Words and phrases which are carelessly selected or carelessly placed produce ambiguous and imprecise specifications.

The main idea behind taking up this work is to propose some techniques / approaches for resolution of ambiguities in SRS document.

1.4 Objectives of the work

The objective of this thesis work is to provide an approach to resolve the ambiguities in the software requirement specifications. The following types of ambiguities are dealt with in this work.

i) Lexical Ambiguity: Ambiguities due to the presence of words having more than one sense/meaning. For example the word ‘Bank’. It has 18 senses according to the WordNet.

ii) Vagueness: ambiguities due to incomplete or vague statements. For example, ‘the temperature should be normal’.

We have used Artificial Immune System (AIS) based Association Rule Mining and Fuzzy Artificial Immune System to resolve the ambiguities present in a SRS document.
1.5 Research Methodology

This work is carried out in two phases. In First phase the study of relevant work has been done to know:

i) What is ambiguity and its types and how an ambiguous SRS document affect the success or failure of a software project?

ii) What is the state of the art in the field?

iii) Which methods can be used to disambiguate an ambiguous statement?

In the second phase a nature-inspired approach is proposed and implemented. The results of the proposed approach are compared with some existing approaches.

1.6 Relevance of the work

The work is for the benefit of Requirement Engineering. As SRS documentation is one of the critical phases of software project development. For the success of a software project it is important that the requirements should be precise and unambiguous.

According to literature survey done, most of the work has been done in the field of ambiguity detection only by using techniques like inspection techniques, checklists etc. some researchers have recommended the use of constraint languages to avoid ambiguity.

The proposed work not just identifies the potential occurrence of ambiguity but provide the best possible sense to the ambiguous word.
1.7 Organization of thesis report

This thesis work is discussed in seven chapters.

**Chapter 2**: it gives the overview about some basic concepts to give background knowledge to the readers. The chapter also discusses the past research, to get the state of the art in the area of disambiguation.

**Chapter 3**: this chapter deals with the study of associative classification. It also discusses the concept and process of association rule mining. Finally it discusses various evaluation strategies.

**Chapter 4**: this chapter explains the concept of Artificial Immune System (AIS), Biological Immune System (BIS) as its background. It also discusses various artificial immune system based algorithms and applications of AIS.

**Chapter 5**: in this chapter, we discuss the proposed system and its implementation.

**Chapter 6**: in this chapter we discuss the results of the proposed system. It also discusses the comparative analysis of the proposed system with some existing approaches.

**Chapter 7**: it concludes the report. Interpretation of results is depicted here and the modifications required in the system are mentioned for future development.

1.8 Conclusion

In this chapter we have discussed how ambiguity plays a vital role in the successful implementation of a software project. The study shows that an ambiguous or inconsistent requirement specification leads to a failed project. Researchers have acknowledged different areas of an SRS document, which are accountable for the success or failure of a software project. Research outcomes reflect that one of the 5 top causes of troubled projects is unclear
or ambiguous requirements. Hence for guarantee the delivery of a quality software project, unambiguous SRS document is an important input.

In general SRS documents are prepared in natural languages so they can be ambiguous as ambiguity is an inherent property of natural languages. This chapter also discusses the objectives of the thesis work and the methodology to achieve the objectives.