Internship Training

At

Design Innovation Center, Jabalpur, M.P.

TECHNIQUES AND TOOLS TO DEAL WITH AMBIGUITIES FOR NATURAL LANGUAGE SOFTWARE REQUIREMENTS.

By

Rashi Singhal

PG/18/058

Under the guidance of

Dr. Nishikant Bele

And

Prof. Dr. S.S. Sandhu

Post Graduate Diploma in Hospital and Health Management

2018-20

International Institute of Health Management Research

New Delhi

COMPLETION OF DISSERTATION FROM RESPECTIVE ORGANIZATION

Design Innovation Center
(Sponsored By MHRD, New Delhi)
Rani Durgavati University,
Jabalpur - 482001(MP)

Prof. Sardul Singh Sandhu
M.Sc., M.Phil., Ph.D., D.Sc. & F.B.S.
Director

सत्यमेव जयते
MHRD

DIC Spokes: IIITDM, Jabalpur: JNKVV, Jabalpur and St. Aloysius Group of Institutes, Jabalpur
DIC Hub: Rani Durgavati University, Jabalpur – 482001 (M.P.) INDIA

COMPLETION OF DISSERTATION FROM RESPECTIVE ORGANIZATION

The certificate is awarded to

Ms. Rashi Singhal

in recognition of having successfully completed his/her Internship in the department of

Title

and has successfully completed his/her Project on

TECHNIQUES AND TOOLS TO DEAL WITH AMBIGUITIES FOR NATURAL LANGUAGE SOFTWARE REQUIREMENTS.

10th May 2020

Design Innovation Center

She/he comes across as a committed, sincere & diligent person who has a strong drive & zeal for learning. We wish him/her all the best for future endeavors.

Sandhu

Training & Development

Sandhu

Zonal Head-Human Resources

Res.: # B-96, Priyadarshini Housing Society, Dumna Road,
Near Hitkarni Engineering College, Jabalpur, India.
Jabalpur, M.P. (482001), India.

Mobile: +91-9424395270,
Land line: +91-761-2602768
Telefax : +91-2608704 (O)

Emails: ssandhu@rediffmail.com, sardulsinghsandhu@gmail.com

<u>TO WHOMSOEVER IT MAY CONCERN</u>

This is to certify that **Ms. Rashi Singhal** student of Post Graduate Diploma in Hospital and Health Management (PGDHM) from International Institute of Health Management Research, New Delhi has undergone internship training at **Design Innovation Center, Jabalpur, M.P.**, from **10th February 2020** to **10th May 2020.** .

The Candidate has successfully carried out the study designated to her during internship training and his/her approach to the study has been sincere, scientific, and analytical.

The internship is in fulfillment of the course requirements.

I wish her all success in all his/her future endeavors.

Dr. Pradeep K Panda	Mentor
Dean Academics and Student Affairs	IIHMR, New Delhi
IIHMR, New Delhi	

<u>CERTIFICATE OF APPROVAL</u>

The following dissertation titled "**TECHNIQUES AND TOOLS TO DEAL WITH AMBIGUITIES FOR NATURAL LANGUAGE SOFTWARE REQUIREMENTS**" at "**Design Innovation Center, Jabalpur, M.P.**" is hereby approved as a certified study in management carried out and presented in a manner satisfactorily to warrant its acceptance as a prerequisite for the award of **Post Graduate Diploma in Health and Hospital Management** for which it has been submitted. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein but approve the dissertation only for the purpose it is submitted.

Dissertation Examination Committee for evaluation of dissertation.

Name	**Signature**
Dr. Madhulika Bhattacharya	
Dr. Sutapa	
Dr. Anandhi	
Dr. Khokkar	

CERTIFICATION FROM DISSERTATION ADVISORY COMMITTEE

This is to certify that **Ms. Rashi Singhal**, a graduate student of the **Post-Graduate Diploma in Health and Hospital Management** has worked under our guidance and supervision. She is submitting this dissertation titled "**TECHNIQUES AND TOOLS TO DEAL WITH AMBIGUITIES FOR NATURAL LANGUAGE SOFTWARE REQUIREMENTS**" at "**Design Innovation Center, Jabalpur, M.P.**" in partial fulfillment of the requirements for the award of the **Post-Graduate Diploma in Health and Hospital Management.**

This dissertation has the requisite standard and to the best of our knowledge no part of it has been reproduced from any other dissertation, monograph, report, or book.

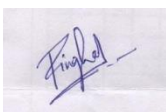
Institute Mentor Name	Organization Mentor Name
Dr. Nishikant Bele	Prof. Dr. S.S. Sandhu
Designation	Designation
Professor	Professor and Dean
Organization	Organization
IIHMR, NEW DELHI	Design Innovation Center,
	Jabalpur, M.P.

INTERNATIONAL INSTITUTE OF HEALTH MANAGEMENT RESEARCH,

NEW DELHI

CERTIFICATE BY SCHOLAR

This is to certify that the dissertation titled "**Techniques and Tools to deal with Ambiguities for Natural Language Software Requirements**" and submitted by **Rashi Singhal** Enrollment No. **PG/18/058** under the supervision of **Dr. Nishikant Bele** for award of Postgraduate Diploma in Hospital and Health Management of the Institute carried out during the period from **10th February 2020** to **10th May 2020** embodies my original work and has not formed the basis for the reward of any degree, diploma associate ship, fellowship, titles in this or any other Institute or other similar institution of higher learning.



Signature

FEEDBACK FORM

Design Innovation Center
(Sponsored By MHRD, New Delhi)
Rani Durgavati University,
Jabalpur - 482001(MP)

Prof. Sardul Singh Sandhu
M.Sc., M.Phil., Ph.D., D.Sc. & F.B.S.
Director

सत्यमेव जयते
MHRD

DIC Spokes: IIITDM, Jabalpur: JNKVV, Jabalpur and St. Aloysius Group of Institutes, Jabalpur
DIC Hub: Rani Durgavati University, Jabalpur – 482001 (M.P.) INDIA

FEEDBACK FORM

Name of the Student: Miss Rashi Singhal

Dissertation Organization: Design Innovation Center, (MHRD), R. D. University Jabalpur.

Area of Dissertation: Techniques and Tools to deal with ambiguities for Natural Language software Requirements.

Objectives achieved: Candidate has achieved success in analysis of ambiguities for Natural Language Software Requirements.

Deliverables: Candidate shown the ability to deal with ambiguities with the techniques and tool studied by her and could be a good model for the analysis.

Strengths: Candidate proved her best ability for the adequate analysis with the available resources and under extreme conditions.

Suggestions for Improvement: More working time could be given for such work could help for a better publication in the Journal of reputes,

Suggestions for Institute (course curriculum, industry interaction, placement, alumni): Institute may improve the course curriculum by more industry oriented projects for better placement and adequate alumni interaction for the placement of the candidates

Sandhu

Signature of the Officer-in-Charge/Organization Mentor (Dissertation)
Date: 15/06/2020
Place: DIC, R.D.University, Jabalpur

Office of Director, Design Innovation Center, MHRD
R.D.V.V. JABALPUR

Res.: # B-96, Priyadarshini Housing Society, Dumna Road,
Near Hitkarni Engineering College, Jabalpur, India.
Jabalpur, M.P. (482001) , India.

Mobile: +91-9424395270,
Land line: +91-761-2602768
Telefax : +91-2608704 (O)

Emails: ssandhu@rediffmail.com, sardulsinghsandhu@gmail.com

ABSTRACT

TECHNIQUES AND TOOLS TO DEAL WITH AMBIGUITIES FOR NATURAL LANGUAGE SOFTWARE REQUIREMENTS.

RASHI SINGHAL

Heath-IT

Keywords: - Disambiguation, Software Requirement Specification (SRS), Natural Language Processing (NLP), Clinical Texts, Clinical Decision Support System (CDSS)
The study aimed at disambiguation of SRS documents with the help of NLP tools and techniques. The study was based on secondary data. The methodology comprised of the tools and techniques available to disambiguate various ambiguities in SRS documents. The applicability of NLP in healthcare industry has been explored with the help of case studies. The fact table summarized types of Lexical, Syntactic and Pragmatic ambiguities and the tools used. The case studies were based on usage of NLP in healthcare industries. The first case study dealt with disambiguation of clinical abbreviations with biomedical texts**.** It has successfully detected duplicity of clinical abbreviations and their varied meanings. However, the tool used by them could handle only the ambiguities in biomedical text. The second case study dealt with handling of the free text of patients' diagnosis and treatment through various NLP approaches in CDSS. Many approaches used in this case study performed well in chunk text classification. They have concluded that there is a need to devise new approaches to deal with clinical document with free text in CDSS**.**
Conclusion- NLP tools are still to be optimized for natural language texts. The transformation of SRS document into a formalized structure is a complex task. QuARS is a good tool for verification and validation of SRS but with the risk of detection of false positives. LOLITA can effectively detect ambiguities in comparison to other NLP tools. The case studies have highlighted the usage of NLP to disambiguate the clinical texts and also how it has impacted the decisions with the help of CDSS.

Acknowledgement

I take this opportunity to express my immense gratitude to various personnel of **Bio design Innovation Center, Jabalpur, M.P.,** and its spoke- **IIITDM, Jabalpur** for providing me a chance to do deep analysis for the technical aspects of requirements gathering process done for software development. I am obliged to **Prof. Dr. S.S. Sandhu**, for accepting me as an intern in their organization. His constant guidance and encouragement helped me immensely in timely completion of my work. Without the constant support of my mentor and guide, **Dr. Nishikant Bele**, I couldn't have completed the study. His exemplary guidance, constant inputs and critical monitoring has not only helped me to understand Health IT better, but also develop my deep analysis skill and provide recommendations for the same.

I am obliged to the Director Sir of my esteemed institution- International Institute of Health Management Research, New Delhi- for providing me this wonderful opportunity to explore myself in organizations like mine. I acknowledge the timely help and suggestions of all the faculty members of IIHMR, New Delhi.

I gratefully acknowledge constant encouragement, motivation and support rendered during the dissertation, especially my parents.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

S. NO.	ABBREVIATIONS	FULL FORM OF ABBREVIATIONS
1.	**SRS**	Software Requirement Specification
2.	**NL**	Natural Language
3.	**NLP**	Natural Language Processing
4.	**WSD**	Word Sense Disambiguation
5.	**ACE**	Attempto Controlled English
6.	**QuARS**	Quality Analyzer for Requirements Specification
7.	**SREE**	Synthesized Requirements Engineering Environment

TECHNIQUES AND TOOLS TO DEAL WITH AMBIGUITIES FOR NATURAL LANGUAGE SOFTWARE REQUIREMENTS

INTRODUCTION

Software Requirements Specification (SRS) refers to documents for software development based on the requirements of the stakeholders, which are prepared in natural language and transformed into a conceptual model. Ambiguities normally creep in SRS because of the usage of Natural Language (NL) in requirements. The NL requirements [**1**]:

a. Needs to shared between stakeholders in designing the software, and

b. Provide inputs as working document for designers, testers and manual editors, who may use it as an agreement document between customers and suppliers or as an information source for project managers **[2].**

There are various reasons for NL to be strongly associated to the requirements analysis [**3**]:

a. **Domain/Scope-** the main challenge for any requirement is to first understand the problem before it is modeled. Therefore, the analyst must interact with users and customers who have different competencies and roles sustained through communication in natural language. The problem becomes acute when there are two sides to be focused on, the first being the approaches requiring modeling of business processes, and the second of progressive integration in information systems of apparatus that does not belong among traditional hardware. Since a change in project is accompanied with corresponding changes in domain of the problem and technical vocabulary, natural language can be used with all interlocutors.

b. **Input-** in majority of cases, it has been noticed that the requirement documents are either available with the help of user or obtained by means of interview of stakeholders. A study concluded that 78% of SRS are embedded in common NL, 16% in structured NL, and only 5% in formal languages [*http://www.online.cs.unitn.it*].

c. **Process**- Software developers consider process as one of the critical activities [*http://www.online.cs.unitn.it*]. It cannot be structured beyond certain limit because it is based on communication with and experience of the analyst. Enabling the user to employ NL not only enhances better understanding of the problem by the analysts but also a better collaboration among the members of a project. It may also facilitate the validation of requirements. Therefore, "customer orientation" is of paramount importance in marketing **[4]**.

d. **Output**- Use of NLP systems has been sought to support the modeling of requirements on the basis of natural language texts **[5, 6].** However, the focus remains on the preliminary analysis of documents to detect ambiguities and signal them to the users or analysts, independently of adopted analysis model. NLP automatic tools have been used to achieve 'disambiguation', which will anticipate validations in the form of narrative descriptions of user requirements.

Requirement engineering ambiguities can be categorized as Lexical, Syntactic, Semantic, Pragmatic, Vagueness, Generality, etc. The identification of ambiguities can be done through Natural Language Processing (NLP). A general use of NL is for requirements understanding to avoid ambiguity and other flaws in the requirements. For example, IBM's Watson **[7]** is proving to be a good tool that can handle NLP, since it can convert text as well context. NLP from SRS to formal language needs to be transformed so that flexibility and simplicity is maintained. This will also bring clarity and completeness in such documents. Despite the advantage, there is a drawback that it might ignore the dynamic aspect of software system. With all the process of transformation taking place, Quality, a major factor for analyzing the requirements in NL cannot be ignored. This will not only help in analysis and transformation of documents, but also helps in verification through formal methods. Automatic quality assessment of the SRS documents should address the reduction in time requirements for key activities and enhance quality in SRS document. The effective solution for the above problem is suggested in following three phases **[8]**:

1. Requirement Gathering & Elicitation phase - automatic quality assessment of the SRS through NLP,
2. Analysis & Specification phase - automatic quality assessment of the SRS driven by NLP, leading to development of conceptual models, and
3. Validation & Feedback phase – extraction of graphical and animated conceptual models from the SRS.

This will provide the writers of the requirements document a tool for alerting them of harmful ambiguities, called nocuous ambiguities (i.e. misinterpretations among stakeholders), in the document **[9].** In fact, the problem of understanding and automatic generation of natural language is closely related to the removal of ambiguities.

My research attempts to find out different ambiguities through various tools employed for NLP in the literature. It has also used a case study based on healthcare to explore disambiguation of NL documents **[10]**. For example, the case study has highlighted that PT meant Patient in the “History of Present Illness” section, Prothrombin Time in the “Labs” section, and Physical Therapy in the “Discharge Instructions” section **[10]**. The exact interpretation of abbreviations has often been approached as a Word Sense Disambiguation (WSD) or text normalization problem **[11, 12, and 13].** Resolving biomedical ambiguities related to abbreviations have been a successful task because of the expansion of the abbreviations in a structure manner. But when it comes to clinical texts, resolving ambiguities has been a tedious task and resulted in failure of NLP techniques as the abbreviations have not been written in the expanded forms.

The case study demonstrates the CDSS based on NLP to extract relevant information from the free text documents **[14]**. Clinical Decision Support System (CDSS) makes use of software designed to help decision making by the clinicians through matching characteristics of individual patients with the computerized knowledge base to facilitate assessments or recommendations specific to patients **[15]**. The CDSS utilizes inputs through structured data, e.g. electronic health records, semi-structured data, e.g. XML documents, and unstructured data, e.g. narrative or free text. Presently, most of the health data of patients is found as unstructured data, adding to ambiguities in clinical decision making. NLP provides appropriate mechanisms to automatically handle the unstructured date by CDSS for arriving at the right decisions.

LITERATURE REVIEW

1. Macias and Pulman **[16]** applied domain- independent NLP techniques to control the production of natural language requirements, proposing the application of NLP techniques to requirements documents in order to control:

i) The vocabulary used

ii) The style of writing

Finally, they discussed as to how NLP techniques can help design subsets of the English grammar to limit the generation of ambiguous sentences.

2. Golden and Berry **[17]** implemented a tool for the extraction of abstractions from natural language texts, i.e. of repeated segments identifying significant concepts on the application field of the problem at hand. They proposed that the technique was restricted to a strict lexical analysis of the text.

3. Hooks **[18]** discussed a set of quality characteristics necessary to produce well-defined natural language requirements.

4. Wilson and others **[19]** evaluated the quality of NL software requirement which could define a quality model that was composed of quality attributes and quality indicators, and thereby developed an automatic tool to perform the analysis against the quality model aiming to detect defects and collect metrics.

5. Fuchs **[20]** proposed to solve problems related to the use of NL in requirements documents by defining a limited natural language, called Attempt Controlled English (ACE). It is easily understood by stakeholders or any other person involved in software development process. It is also simple enough to avoid ambiguities allowing domain specialists to express requirements using natural language expressions and to combine these with the rigour of formal specification languages.

6. Kamsties and Paech **[21]** concluded that ambiguity in requirement was not just a linguistic specific problem and hence proposed the idea of a checklist that could address not only linguistic ambiguity but also ambiguity related to a particular domain.

7. Mich and Garigliano **[22]** proposed a set of measures for semantic and syntactic ambiguity in requirements. Their approach was based on the use of information on the possible meanings and roles of the words within a sentence as well as the possible interpretation of a sentence. This was achieved by using the functionalities of a tool called LOLITA.

8. Natt och Dag et. al. **[23]** presented an approach based on statistical techniques for the similarity analysis of NL requirements aimed at identifying duplicate requirement pairs. It may be used successfully for revealing inter-dependencies and then may be used as a support for the consistency analysis of NL requirements.

9. Ambriola and Gervasi **[24]** developed a web-based NLP tool, called Circe, designed to facilitate the gathering, elicitation, selection, and validation of NL requirements.

10. **IBM Rational Doors,** a requirements management tool, provides relevant functional modules for the generation of NL requirements and the traceability among NL requirements [*http://www-01.ibm.com/software/awdtools/doors/*].

11. Goldin and Berry **[25]** implemented a tool, called Abstfinder, to identify the abstractions from natural language text used for requirements elicitation.

12. Lee and Bryant **[26]** developed an automated system to assist the engineers to build a formal representation from informal requirements like NL requirements.

13. Kamsties et al. **[27]** described a pattern-driven inspection technique to detect ambiguities in NL requirements.

14. Fuchs and Swhwitter **[28]** present a restricted NL, called Attempt Controlled English (ACE), to translate specifications into sentences in first-order logic in order to reduce ambiguity in requirement specifications.

15. Mich and Garigliano **[29]** explored the use of a set of ambiguity indices for the measurement in syntactic and semantic ambiguity, which was implemented using an NLP system called LOLITA.

16. Kiyavitskaya et al. **[30]** proposed a two-step approach where a set of lexical and syntactic ambiguity measures were firstly applied to ambiguity identification, followed by a tool measuring potentially ambiguity specific to each sentence.

17. Many researchers have focused on either corpus-based statistical methods or linguistic approaches. The latter used part-of-speech (POS) tagging and shallow and deep parsing information to apply pattern- or rule- based matching **[28, 31, and 32].**

18. Resnik **[33]** took advantage of semantic similarity of taxonomy to resolve coordination ambiguity involving nominal compounds.

19. The use of manual inspection was and is still the most popular way to detect and resolve ambiguities. Since the natural language requirements specifications are inherently ambiguous, the use of formal specifications was absolutely necessary to resolve these ambiguities **[34]**. Meyer's approach to detecting ambiguities was to inspect each word, phrase and sentence manually. Kamsties et al. **[35]** proposed a specific methodology of human inspection to resolve ambiguity. Letier et al. **[36]** proposed the use of formal specifications to validate requirements.

20. Ambriola et al. **[37]** attempted to validate NL Specification with the help of the user after deriving a conceptual model automatically from the requirements specifications using the tool, called “Circe”, which was funded by IBM and now available as plug-in for “Eclipse”.

21. Many studies attempted to reduce the problems associated with unrestricted NL by limiting the scope of the language, and restricting the grammar to consider only a subset of NL when writing a requirement specification **[38, 39, 40, and 41].**

22. Osborne et al. **[42]** dealt with unrestricted language by using techniques developed in NL processing (NLP) in order to detect ambiguities in SRS documents through syntax.

23. Another interesting tool was that of Wilson et al. **[43, 44],** which used nine quality indicators for requirements specification, viz. Imperatives, Continuances, Directives, Options, Size, Specification Depth, Readability, Text Structure and Weak Phrases. Their results showed only the frequency counts of these indicators in different samples, without taking the crucial decision of whether or not a sample is ambiguous.

24. A tool, called “QuARS” (Quality Analyzer for Requirements Specification) **[45, 46]**, syntactically parsed the sentences using the MINIPAR parser **[47],** and then combining both lexical and syntactic information to detect specific ambiguity indicators of poor-quality requirements specification.

<u>METHODOLOGY</u>

Key research question- How to disambiguate **SRS** documents through NLP tools and techniques.

Research design- There is various tools and techniques of NLP available to ensure that the ambiguities are removed from the SRS document. Few tools can be categorized as:

1. For the removal of semantic and syntactic ambiguities, **SREE** (Synthesized Requirements Engineering Environment) tool has been used **[48, 49]**.

2. **ACE** (Attempto Controlled English) **[50]** and **ANLT** (Alvey Natural Language Toolkit) **[51] have been used as** advance tools for automatic semantic analysis and controlled language (i.e. fixed vocabulary), respectively.

3. A domain specific **RESI** (Requirement Engineering Specification Improver) **[52]** tool has also been used. The word ontology **[53, 54, 55, 56, 57, 58, 59, 60, and 61]** attempted to provide unambiguous protocols for communication, portability, interoperability and reusability. **NlrpBench [62]** is helping in comparative analysis of the different tools of NLP**. SBVR** (Semantics of Business Vocabulary and business Rules) has been attempted for disambiguation **[63, 64]**. **CKCO** (Context Knowledge and Concepts Ontology) **[65]** attempted to resolve lexical-semantic ambiguity in natural language.

4. Parsing technique or syntactic analysis can be done on sentences using **REED-KELLOG** sentence diagramming system **[66]** to obtain a simple structure of a sentence.

5. **QuARS** (Quality Analyzer for Requirement Specifications) was employed for analysis of quality model and real requirements **[1]**. This aimed to perform

quantitative (allowing metrics collection), corrective (enabling detection and correction of defects) and repeatable (which ensuring a similar output for a similar input across domains) evaluations.

6. To find out the nocuous ambiguities, **NAI** (Nocuous Ambiguity Identifier) **[67]** was used. A part of the tool **NER** (Named Entity Recognition) enabled extraction of coordination constituents from the sentences.
7. **LOLITA [68, 69, 70]**, a system, analyzed different ambiguity measures for semantic as well as syntactic ambiguities.

All the tools of NLP mentioned above can be compared with the help of nlrpBENCH. This will help in making up the culture of collaboration, openness and publicness in future.

Research procedures- The procedures for disambiguation can be understood at two levels: surface understanding (i.e. literal) and conceptual understanding (i.e. modeling). This revolves around three basic parameters of NLP that plays very important role for the procedures to take place, viz. Precision, Recall, and F-measure. **Precision** means the proportion of the relevant results. **Recall** denotes the proportion of the correct results. **F-measure** refers to weighted harmonic mean of the precision and recall of the test. Some of the major steps to transform the unstructured SRS document into a formalized or structured document with less ambiguities using NLP are summarized as below:

1. **Text Pre-processing Module [79] -** helps to identify the ambiguous sentences using NLP that extracts requirements from documents, does machine translation, and extracts ambiguous requirements. **Hybrid approach** is preferred for text pre-processing, consisting of **Rule based approach** (i.e. to remove ambiguity, but not

sufficient due to the requirement of context knowledge) and **Statistical based approach** (i.e. when data is voluminous).

The sub-parts of text pre-processing module include:

a. **Tokenization-** means breaking up the input into tokens, i.e. a word, a number, or a symbol.

b. **Sentence Splitting-** splits the text into sentences.

c. **Part-of-Speech (POS) Tagging-** marks every word of a sentence with pre-defined parts-of-speech, **i.e.** each token being annotated with a POS tag including adjective, adverb, noun, or verb **[71, 72].**

d. **Named Entity Recognition (NER) -** identifies organizations and locations including domain keywords and component names.

2. **Parsing (Syntactic analysis or Text chunking) [1] -** helps to structure the sentences by removing the lexical and syntactic ambiguities, ensuring a proper understanding of requirements between the stakeholders or users. REED-KELLOG **[66]** parsing technique can be used with the help of the particular schemata suggested: Sentence ⟵ subject + predicate (verb + object), and will be filtered down to constrained natural language as: Requirement ⟵ subject + verb + target + [way].

Most commonly, **Stanford Parser** has been used for parsing the sentences, but it could provide wrong results owing to errors of syntactic ambiguity. During the parsing activity, there are some quality properties evaluated using indicators for detecting and measuring the requirements document **[Table 1]**, such as

i. **Non-ambiguity**- where the indicators are vagueness, subjectivity, optionality, weakness.

ii. **Specification completion**- where the indicator is under-specification.

iii. **Consistency**- where the indicator is under-reference.

iv. **Understandability**- where the indicators are multiplicity, implicitly, un-explanation.

3. **QuARS [1]-** is sufficient for comparison and verification of the quality of SRS and the indicators are sufficient to be included as part of syntax and structural related issues. There are main logical modules for this tool to work efficiently, namely, Lexical analyzer, Syntax analyzer, Quality evaluator, Specific purpose grammar, Dictionaries **[Fig. 1]**.

Table 1: Properties and indicators for detecting and measuring the requirement document [1].

PROP.	*INDICATOR*	*DESCRIPTION*	*N(*
NON-AMBIGUITY	Vagueness	A Vagueness Indicator is pointed out if the sentence includes words holding inherent vagueness, i.e. words having a non uniquely quantifiable meaning	Vagueness-revealin *strong, good, bad, adequate, recent, ..*
	Subjectivity	A Subjectivity Indicator is pointed out if sentence refers to personal opinions or feeling	Subjectivity-reveali *similarly, having ir account, as [adjecti*
	Optionality	An Optionality Indicator reveals a requirement sentence containing an optional part (i.e. a part that can or cannot considered)	Optionality-revealir *eventually, if case, i appropriate, if need*
	Weakness	A Weakness Indicator is pointed out in a sentence when it contains a weak main verb	Weak verbs*: could,*
SPECIFICATION COMPLETION	Under-specification	An Under-specification Indicator is pointed out in a sentence when the subject of the sentence contains a word identifying a class of objects without a specifyer of this class	Words needing to b flow (data flow, cor (write access, remo access, ..), testing (f structural testing, ur
CY	Under-reference	An Under-reference Indicator is pointed out in a RSD (Requirement Specifications	-

PROP.	INDICATOR	DESCRIPTION	N
UNDERSTANDABILITY	Multiplicity	A Multiplicity Indicator is pointed out in a sentence if the sentence has more than one main verb or more than one direct or indirect complement that specifies its subject	Multiplicity-reveal *and/or, ...*
	Implicity	An Implicity Indicator is pointed out in a sentence when the subject is generic rather than specific.	Subject expressed l Demonstrative adje *that, those*) or Pron Subject specified b Adjective (*previou last...)* or Prepositi
	Unexplana-tion	An Unexplanation Indicator is pointed out in a RSD (Requirement Specifications	-

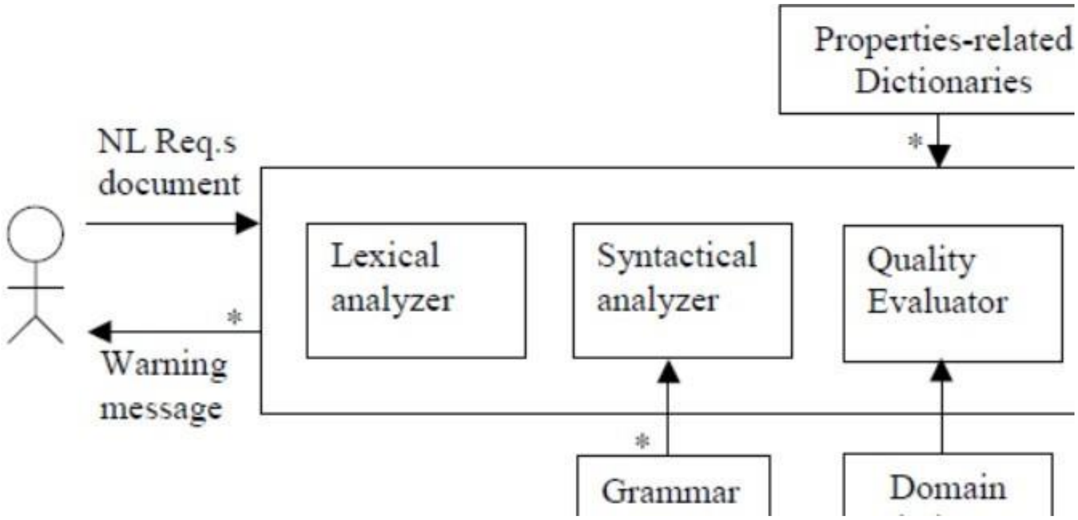
QuARS follows a specific procedure to analyze the requirements in a document.

The steps are:

a. **Lexical Analyzer** – verifies whether a correct English Dictionary has been used in the SRS document or not.

b. **Syntactical Analyzer** - uses a **special purpose grammar** and builds the derivation trees of each sentence from the output of the first step.

c. **Quality Evaluator module** - receives **properties-related and domain dictionaries** as input, containing the words and the syntactical elements, facilitating detection of inaccuracies in the SRS. This may warn the users of potential defects in the SRS document.



4. **Nocuous ambiguity classification module (heuristics to predict nocuity) -** helps to identify nocuous ambiguities based on high and low attachments interpretations. Three major thresholds for this module are **[67]:**

   a. **Building dataset-** is based on collection of ambiguous sentences by humans as well as using the heuristics to replicate the human judgments.

   b. **Training classifier-** measures the agreement needed from the judges over a particular interpretation, permitting adjustment in tolerance levels. The interpretation certainty is quantified as proportionate judgments against the total judgments for the whole instances. If the attachments have certainty greater than ambiguity threshold, implying that the coordination instances are innocuous ambiguity.

   c. **Applying the classifier-** a feature vector in the classifier allowing the return of the predicted class label by the classifier.

The above procedures have helped in finding out the harmful ambiguities and reducing them to certain extent.

This procedure is explained with the help of a case study on disambiguation of clinical abbreviations with biomedical texts **[10].** NLP for biomedical texts have inherent difficulties in disambiguating abbreviations of the clinical texts. The methodology used in the case study **[10]** detected abbreviations and attempted to disambiguate them along with their evaluation. It involved detection of the abbreviations and choosing their correct form. The tools for NLP applications such as Metamap **[73]** and POET (Parsable Output Extracted from Text) were experimented on biomedical and clinical texts at the University of Utah, U.S.A., who used ABRADe (Abbreviations and Acronyms Disambiguation) as an evaluation tool. They also used SPECIALIST Lexicon LRABR resource as a dictionary lookup module for abbreviations detection. Two approaches were employed to disambiguate abbreviations, namely, LIBLINEAR as semi-supervised multi class SVM classifier [**74**] and Ling Pipe as an unsupervised clustering method [**75**]. The outputs by LIBLINEAR were validated through 5-fold cross validation on the training cases. The correct form of abbreviations was chosen by machine learning approaches, requiring more domain specific knowledge like the UMLS SPECIALIST lexicon [**76**]. The latter is a large biomedical and general English syntactic lexicon and developed by the National Library of Medicine at NIH, Maryland, U.S.A. to provide the lexicon information needed by various NLP applications.

The abbreviations were detected with pre-processing of documents. The sentences were split by a splitter, adapted from openNLP, and the cTAKES **[77]** trained model, and a tokenizer. All the abbreviations were annotated by pattern matching and LRABR, and finally disambiguated. They evaluated the detection of abbreviations with a reference standard of 37 randomly selected clinical notes that were manually annotated with disagreement adjudication. For the semi-supervised approach, the remaining 9,963 randomly selected clinical notes created “synthetic” training cases using a method similar to that by Pakhomov **[78].** LRABR detected all the expanded forms (e.g. cerebrovascular accident) in 9,963 clinical notes, and then replaced them with the corresponding abbreviation (e.g. “CVA”).

CDSS based on NLP helps various stakeholders of healthcare system by supporting diagnosis, cares, screening, treatments, tracking, and monitoring [Fig. 2].

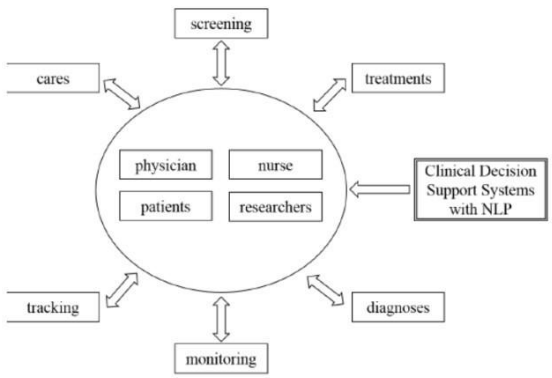


Figure 2: CDSS based NLP in healthcare management.

The case study reviewed different approaches used in CDSS based on NLP and analyzed the following features analyzed:

a. **Language**- determines the appropriate approach of NLP for free texts, many approaches have considered English as their domain language and few approaches not using English as domain language **[79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, and 91]**.

b. **Document type with free text as input**- NLP techniques have been proposed to process a variety of clinical documents such as radiology reports, narrative reports, notes, or texts, progress notes, patient questionnaire, lab reports, chest X-ray and radiography reports, pathology reports, and electronic health records.

c. **Level of NLP-** Domain language approaches were handled through syntactic analysis, semantic analysis, pattern based approaches or rule-based on regular expressions, keyword searches, machine learning, and heuristic method. Metamap has been used to remove all negated concepts (using NegEx), semantic type filtering semantic-type finding and filtering of high-level concepts **[90]**.

d. **Patient type-** Patients have been classified as inpatients or outpatients, receiving direct and indirect effects of a clinical decision making by physicians based on monitoring the symptoms, performing the diagnostics, applying the medical treatment, and recommending intensive care.

e. **Clinical decision support task-** Various NLP tasks identified of risk factors including adverse clinical events and the findings and concepts generally associated with diseases like pneumonia, extracted problem list and medical problems, labeled patients based on their personal details, and detected explicit guidelines.

f. **Health outcomes- CDSS based on** NLP analyzed health outcomes of various approaches and its effect on health professionals and patients.

RESULTS

The results section highlights two major aspects, which are:

1. The fact tables describing the classification and sub-classification of ambiguities and the relevant tools and techniques helping to resolve such ambiguities.
2. A case study related to healthcare industry focused on reduction of the ambiguities in the clinical abbreviations when compared to the biomedical texts by NLP tools **[10]**.

1. There is no clear demarcation among the ambiguities mentioned in the literature. Therefore, after a thorough literature review, an effort to distinguish the types and sub-types of ambiguities has been made and appended with the relevant tools and techniques used for resolving ambiguities.

Lexical Ambiguity - It is an error that occurs when a word in a sentence or the whole sentence has multiple meanings. Table 2 enlists sub-types of lexical ambiguity and the tools and techniques applied for their reduction or elimination.

Table 2: Sub-types of Lexical Ambiguity.

Sub-type of Ambiguity	**Tools & Techniques**	**Reference**
1. **Homonym-** the words with similar written and phonetic representations with dissimilar meanings and different etymologies. 2. **Polysemy-** the words with multiple related meanings but one etymology. 3. **Semantic-** A sentence having more than one way of reading within its context. It can be caused by other	**NAI** with the classifier.	**[67]**
	QuARS	**[1]**
	Checklist based Inspection- not efficient	**[92], [98]**
	Style Guides- not capable to deal with vagueness, plural noun ambiguity, and complex coordination ambiguity.	**[92]**
	SREE	**[92]**

<table>
<tr><th>Sub-type of Ambiguity</th><th>Tools & Techniques</th><th>Reference</th></tr>
<tr><td rowspan="4">ambiguities like:
(i) Coordination ambiguity (discussed later)
(ii) Referential ambiguity (discussed later)
(iii) Scope ambiguity- includes some quantifier operators like a, all, each, every, some, several with not as negation operators.</td><td>Controlled Language-
a. AMBIDEXTER is based on context free grammar.
b. ACE with a restricted grammar and domain-specific vocabulary.
c. ANLT</td><td>[109]

[50]
[51]</td></tr>
<tr><td>Knowledge based Approach-
a. RESI
b. SBVR
c. CKCO</td><td>[52]
[63, 64]
[65]</td></tr>
<tr><td>Heuristics Based Approach- machine learning approaches.
a. NAI</td><td>[67]</td></tr>
<tr><td>Statistical Based- Tree Tagger, Dowser and Stanford Parser</td><td>[92]</td></tr>
</table>

Structural/ Syntactic Ambiguity - It refers to multiple grammatical structure to sequence of words, having different meanings **[Table 3]**.

Pragmatic Ambiguity - This type of ambiguity refers to the study of relations between language and context, having linguistic and philosophical roots. It is concerned with context dependent meaning **[Table 4]**.

Table 3: Sub-types of Syntactic Ambiguity.

Sub-type of Ambiguity	Tools & Techniques	Reference
1. **Analytical ambiguity-** when a noun group including modifier scope creates ambiguity in the role of constituents within a phrase or sentence. 2. **Attachment ambiguity-** when prepositional phrase or a relative clause can be legally attached to two parts of a sentence. 3. **Coordination ambiguity-** it occurs: i. When **and** or **or** is used in a sentence, or ii.When a conjunction is used with a modifier.	**Human Judgments** **Heuristic Approach-** Statistical method**- BNC, using Sketch Engine. Collocation frequency** and semantic similarity. **NAI** is also used for coordination ambiguity**.** **Machine Learning Approach-** decision tree, Logistic Regression, **LogitBoost for "nocuity classifier".** Tool Used for identifying coordination ambiguity**- Named Entity Recognition (NER).** To validate or verify the quality of requirements document **QuARS** tool is being used.	**[67], [98], [92],**

Table 4: Sub-types of Pragmatic Ambiguity.

Sub-type of Ambiguity	Tools & Techniques	Reference
Referential Ambiguity- is an intermediate of semantic and pragmatic ambiguity, since it may take place within a sentence or between a sentence and its discourse context. However, it is more inclined towards pragmatic ambiguity.	**ANLT**	**[51], [98]**

2. The results of all the three broad categories of ambiguities are assessed with the help of three major statistical factors, namely, **Precision, Recall, and F –measure [10].** Therefore, the case study also portrays its result in the form of the three major statistical factors for three major approaches, namely, detection of abbreviations, semi-supervised abbreviations disambiguation and finally the unsupervised abbreviations disambiguation.

<u>**Detection of Abbreviations**</u>- A satisfactory performance was observed for abbreviation detection, but when the references were compared with all the abbreviations detected by the system it did not perform excellently. Even the

partial matches could attain only 75% F-measure [**Table 5**]. This has highlighted the limitation of using the biomedical text resource with clinical abbreviations, since LRABR could detect only the biomedical abbreviations. A list of false negatives (i.e. the most frequently missed) and false positives (i.e. the spurious) abbreviations highlighted the instances in 37 documents reference standard [**Table 6**].

TABLE 5: Abbreviation detection evaluation results of the case study [10].

	Recall	Precision	F1·
Exact match	63.71	68.32	
[illegible]	[illegible]	[illegible]	

TABLE 6:Top false negative and false positive abbreviations in the case study [10].

False negatives		False posi	
Abbreviation	Instances	Abbreviation	
p.o.	48	He	
b.i.d.	27	q	
q.d.	21	b	
MedQ	15	In	
Mrs.	13	At	
HEENT	10	K	
mL	8	C	
[illegible]	[illegible]	[illegible]	

Spurious abbreviations were listed in LRABR and not in the reference standard. For example, “He”, commonly used as pronoun, was listed as an abbreviation of Helium by the LRABR and detected so by the system [**10**].

<u>Semi-supervised abbreviations disambiguation</u>- From the previous work of finding out the ambiguous abbreviations from 10,000 notes, currently 12 most frequent ambiguous abbreviations in 9,963 training set notes were selected from ten different types of notes (viz. Social Service Note IP, Rheumatology

Clinic Note, Plastic Surgery Clinic Note, Operative Report, Obstetrics Gynecology Clinic Note, Hematology Oncology Clinic Note, Discharge Summary, Cardiology Clinic Note, Burn Clinic Note, and Admission H&P). When LRABR was being used for abbreviation and its expansion, 22,822 instances were extracted with dissimilar distributions of each abbreviation [**Table 7 and 8**].

TABLE 7: Instances of abbreviations in the training corpus with diverse expanded forms in the case study [10].

Abbreviation	Instances in corpus	Different expanded forms in LRABR	Different expanded forms in corpus	Usage of LRABR (%)
PO	9,030	10	4	40.00
CA	8,390	66	22	33.33
AP	3,893	47	21	44.68
ER	1,530	21	5	23.81
RA	1,121	27	14	51.85
LV	698	13	8	61.54
IV	392	16	6	37.50
DVT	310	2	2	100.00
MD	163	38	16	42.11
PND	134	9	3	33.33
CVA	113	3	3	100.00
CT	48	4	4	100.00

Table 8: Results of the semi-supervised method used in the case study [10].

Abbreviation	LIBLINEAR accuracy (%)	The most common (%)	c
PO	97.29	49.86	
CA	89.58	58.95	
AP	86.62	43.98	
ER	96.21	65.23	
RA	77.07	33.81	
LV	90.69	80.95	
IV	83.42	71.94	
DVT	68.71	75.16	
MD	46.01	39.26	
PND	89.55	89.55	
CVA	98.23	61.06	

Unsupervised abbreviations clustering- The training instances were also tested by using the clustering algorithm. The obtained clusters were compared by LingPipe with those used for semi-supervised method as reference standard. Recall and precision in clustering were quantified by comparing instances with the reference standard in each cluster, and counting them as true positive [**Table 9**]. A different expanded form compared to the one assigned to cluster constituted the false positive. Those not found in the cluster with the same expanded form constituted the false negative. The study failed to produce true negatives since accuracy and F-measure were equivalent in the semi- supervised method, requiring prediction of at least one expanded form by the classifier. Generally, the results obtained by the unsupervised method had lower accuracy than that by the semi-supervised results **[10].**

TABLE 9: Results of expanded form clustering in the case study [10]. (Difference = LIBLINEAR – F1-measure)

Abbreviation	Clustering			LIBLINEAR accuracy (%)	Difference
	Recall	Precision	F_1-measure		
PO	99.93	48.57	65.37	97.29	31.92
CA	99.64	39.77	56.85	89.58	32.73
AP	98.94	26.76	42.12	86.62	44.50
ER	99.42	50.88	67.31	96.21	28.90
RA	97.24	21.89	35.74	77.07	41.33
LV	98.89	67.45	80.20	90.69	10.49
IV	97.80	54.11	69.68	83.42	13.74
DVT	99.75	62.91	77.16	68.71	-8.45
MD	82.33	20.70	33.08	46.01	12.93
PND	96.75	80.89	88.11	89.55	1.44
CVA	98.73	52.97	68.94	98.23	29.29
CT	90.56	44.48	59.66	68.75	9.09
Micro Avg.	**99.33**	**42.17**	**58.45**	**91.09**	**32.64**
Macro Avg.	**96.67**	**47.62**	**62.02**	**82.68**	**20.66**

NLP approaches were evaluated with the average classification, precision and recall. Medical Detection Identification System (MeDS) and rule based approach for screening the cervical cancer have recorded good performance for chunk text classification with an average rating of 96% and 98%, respectively [Table 1]**[82, 83]**. MediClass approach performed poorly with average classification of only 57%, owing to the implementation of NLP technique with a shallow textual analysis. An analysis conducted by precision, recall, and F-measure of the approaches performing the NLP task called information extraction (IE) from free text. All approaches have recorded better results in recall than precision due to the extraction of irrelevant information. Various approaches for CDSS with NLP were performed to expose an updated state of the art. The six valuable features selected provided and facilitated the selection of the most suitable approach to solve problems related to CDS.

<u>DISCUSSION</u>

Disambiguation of SRS document is not an easy task as it involves complex procedures to be performed by the experts. It is not possible to automate the complete process since the experts are required to solve complex ambiguities with the domain knowledge inherent. Controlled language is one of the good ways to disambiguate the SRS document but with the limitation of focusing on only two ambiguities, i.e. Lexical and Structural/Syntactic. Knowledge based approach, machine learning and ontology approaches are able to give precise results by identifying semantic ambiguities from requirement specification [**92**]. The NLP tools help to transform the unstructured SRS document into structured format, i.e. into a more formalized manner. But some of the tools mentioned helps only in transforming the document into one specific format rather into any format required. **LOLITA** is the biggest NLP system used for disambiguation with the help of ambiguity measures **[68, 69, and 70]**. The major evaluations of the LOLITA are discussed below:

a. **Semantic ambiguity-** LOLITA consists of the larger semantic net in comparison to other NLP systems. The main advantage is that it can be used to obtain the ambiguity of terms within the individual parsing trees [**3**].

b. **Syntactic Ambiguity- As** there is a difficulty in ordering the parsing trees, the correct interpretations are not shown in first positions. An analysis conducted in this regard yielded 56% in the first three positions **[3]** [*Better results are expected from the new version of LOLITA, **Concepts**, which is about to be marketed*]. It requires more efforts by the analysts and makes it essential, so that results of analysis are presented in a way that facilitates the identification of ambiguity.

c. **Evaluation of parsing with LOLITA-** data are available which have been obtained from analysis of texts of differing quality and therefore characterized by various levels of ambiguity **[3]**, and where 20% of the sentences have only 1 parse tree, 25% have 2-9 parses, and the rest have more (i.e. 3% no parses and 8% timed out). It helped in establishing an acceptable ambiguity threshold that was based on the importance of the documents concerned, and which in the case of requirements need not be absolute.

Another tool that can be considered important in classifying the ambiguities as nocuous or innocuous is the **NAI** with the classifier being a sub-part of the tool. This tool used the heuristic approach using the classifier to distinguish between nocuous and innocuous ambiguities. The classifier used the LogitBoost (LB) algorithm that was based on machine learning approach **[67]**. With this approach, various ambiguity thresholds were set and compared with the baselines. In comparison to the precision baseline (P_BL), LogitBoost classifier performed with precision of up to 75% on average at different threshold level **[67]**. LogitBoost could successfully replace the original regression (LR) algorithm. The major evaluation on the basis of baselines could be stated as, most of the instances at low ambiguity thresholds were judged as innocuous ambiguity and the instances at high ambiguity thresholds were judged as nocuous ambiguity.

Apart from the systems being used for NLP, templates are another way or tool for reducing ambiguities in NL requirements. It makes the requirements friendlier to automated analysis **[92, 93, 94, and 95].** Templates are also known as boilerplates, melds, and patterns **[96, 97].** The syntactic structure of requirements can be put into number of pre-defined slots. To verify that requirements conform to templates, quality assurance task was used **[99].** When requirements domain keywords or glossary terms were known, automation of conformance checking to templates was done with relative ease. In the literature, two main templates have been mentioned, namely, Rupp's Template **[94]** and EARS template **[100, 101, 102].** RETA (REquirments Template Analyzer) has helped to enable the analysts to automatically check conformance to both the templates **[99]**. This will also help to find out errors in structural analysis of requirements **[99].** In comparison to Rupp's template, EARS has permitted advanced features for specifying conditions. The major limitation of templates relates to the glossary because some experts refer to it after the conformance checking. Besides, glossary might not contain all the major keywords at the time of conformance checking. Therefore, an attempt to conformance checking of requirements to the templates should be done without much dependency on the glossary.

NLP tools and techniques are not always effective and efficient in usage. It needs to be validated and verified with the help of the domain experts who will provide their own judgments with the help of the automated tools and techniques. For the very same purpose, i.e. to verify and validate, a tool called QuARS has been used, which is comparatively effective and efficient in comparison to other tools **[1]**. QuARS aims to match some major characteristics like

a. **Ease to use**- The people are easily trained improving time logistics, which was matched by the TCL/TK graphical interface **[103]**.

b. **Generality**- The evaluation of the text format allows high generality since each electronic format may always be saved as text format.

c. **Flexibility**- QuARS allows evolution and modification of dictionaries enabling detection of various indicators.

d. **Multi-lingual-** QuARS can analyze requirements in multiple languages by changing the lexical and syntactical analyzers, and translate the dictionaries in a new target language.

After in-depth evaluations on the types of ambiguities and the major tools and techniques used to address them, a case study involving disambiguation of clinical abbreviation with biomedical texts was used to test their efficacy **[10]**. The methodology experimented in the case study to disambiguate the text of biomedical with clinical data could not perform well owing to the lack of **Domain or institution specific knowledge,** which, if embedded, may increase recall significantly. The probable approaches for realizing improvements in the resources used may be diverse. False positives can be minimized with some specific keywords like “He”, “In”, or “At”. Recall may be improved by removing case information or punctuation in abbreviations. For example, “mL” usually means “milliliter” and “dl” for “deciliter”, but their abbreviations are “ml” and “dl” in LRABR. Recall can be further improved by detecting abbreviations with an edit distance method [**104**] instead of strict string matching. Normalizing these terms could further improve the performance with the help of UMLS Metathesaurus, which could be used for linking all the synonyms. The unsupervised abbreviations clustering method gave encouraging results and can be improved in many ways like a wider context window, using whole words in a sentence, using surrounding sentences as features, etc. If the two approaches (i.e. semi-supervised and unsupervised) were compared, the only difference between them was the hypothesis of collocation proposed by Yarowsky [**105**], which meant that the unsupervised abbreviations approach did not depend on the “one sense per collocation”, whereas the other approach did. The methodology used in the case study was not the only work on the disambiguation of abbreviations in clinical text. Rather, there are many researchers who have contributed to the on the disambiguation of abbreviations in clinical text. WSD is the complex procedure that is performed in the healthcare industry. Since the database of the industry has expanded from EMR's to EHR's becoming voluminous, it has become difficult to disambiguate the text abbreviation or notes easily. Since clinical notes are based on medical knowledge, biomedical and clinical domain resources, it can therefore serve as the knowledge base to enhance clinical WSD algorithms. In particular, UMLS and the SNOMED-CT (Systematized Nomenclature of Medicine-Clinical Terms) could be utilized as terminology resources in the biomedical and clinical domains, respectively. They may prove to be of great help in disambiguating the clinical texts for its processing by NL techniques, which can immensely improve time, treatment and training logistics in health care.

Language- There is a lot of complexity in the usage of various natural languages in clinical documents. CDSS supports only the English that is the most common natural language used in healthcare domain.

Document with free text- NLP systems yield better results with structured data in clinical documents but fail to do so with free texts. Efficient NLP approaches are, therefore, required to process clinical free text in efficient administration of healthcare.

Level of NLP approach- NLP approaches faced greater challenges in deep linguistic analysis since it is required to understand natural language phenomena and extract hidden information in clinical documents.

Patient type, task, and health outcomes- NLP approaches in CDSS can help the patients with efficient identification of adverse events, unchained disease symptoms, emotions or risk factors, monitoring treatment, or clinical follow-up.

LIMITATIONS

Despite many advancements and advantages, NLP techniques and tools pose major limitations in healthcare industry, such as

a. Lack of clear cut distinction in the types of ambiguities in SRS document.
b. The major focus has been on only lexical and syntactic ambiguities.
c. Pragmatic ambiguity has largely remained untouched since it is complex to deal with.
d. NLP tools and techniques are still not optimized creating lot of ambiguities in requirements documents.
e. NLP tools still fail to efficiently deal with healthcare requirement documents.
f. There is not much literature available on resolving the ambiguities associated with processing of SRS healthcare documents.
g. The sub-types of ambiguities could not be directly addressed in the dissertation owing to paucity of relevant literature.

CONCLUSION

The paper documents various ambiguities in SRS documents and suggests their plausible resolutions through diverse tools and techniques. The NLP tools using knowledge based ontology and machine learning approaches are efficient enough to find out semantic ambiguity. However, majority NLP tools are still to be optimized with respect to pragmatic and syntactic ambiguities. Since nlrpBENCH may set standard for NLP tools, both professionals and researchers need to use, expand and improve nlrpBENCH. Presently, the transformation of SRS document into a formalized structure is a complex task and therefore, many industries still prefer informal representation of requirements. However, formalization of the SRS is being attempted across industries especially the healthcare by using the conceptual models. QuARS has proven to be a good tool for verification and validation of SRS but with a limitation that it might sometimes highlight the risk of detection of false positives. This can be looked after with the help of dictionaries or glossary already embedded in the NLP systems. To detect the ambiguities at an early stage of software development, the automated techniques must be scalable to point out whether they are nocuous or innocuous. NAI has proven itself as a good tool until now but it still needs to be further optimized. Another system, LOLITA, can effectively detect ambiguities in comparison to other NLP tools. To disambiguate a text, a hierarchical quality model was developed in the literature and applied to the healthcare industry **[10]**. The case studies mentioned have highlighted the usage of NLP to disambiguate the clinical texts and also how it has impacted the decisions with the help of CDSS. Therefore, despite many researchers going against NLP, it is being preferred by many for disambiguating the SRS documents. The second case study **[14]** highlighted that the free clinical texts are still a challenge for NLP applications in clinical decision support system.

FUTURE WORK

a. The conceptual framework of SRS still requires lot of work by the researchers and users, especially in healthcare industry.
b. SRS needs to be transformed into formal language, i.e. may be object oriented model.
c. More heuristic approaches are to be developed for extraction of ambiguities, improving the accuracy of the tools.
d. QuARS can be used not only for validating SRS ambiguities but also to validate and verify the checklists, questionnaires and user manuals. The quality model can be further expanded and refined in order to provide support completely.

e. Researches should be initiated to identify objective indicators for SRS conceptual understanding criteria, ensuring that ambiguities are not the only cause of misunderstandings.

RECOMMENDATIONS

1. A standard language format should be developed for use by medical professionals in order to reduce ambiguities during requirement processing.
2. More NLP tools should be employed in healthcare industry.
3. Validation and verification of SRS using NLP tools must be focused upon.

BIBLIOGRAPHY

1. Fabbrini, F., Fusani, M., Gnesi, S., & Lami, G. (2001). The linguistic approach to the natural language requirements quality: Benefit of the use of an automatic tool. *26th Annual NASA Goddard Software Engineering Workshop, IEEE/NASA SEW 2001*, 97–105. https://doi.org/10.1109/SEW.2001.992662 .

2. N. Power Variety and Quality in Requirements Documentation Seventh International Workshop on Requirements Engineering: Foundation for Software Quality, Interlaken, Switzerland, June 4-5 2001.

3. Mich, L., & Garigliano, R. (2000). Ambiguity measures in requirement engineering. *Proceedings of International Conference on Software—Theory and Practice (ICS2000), Sixteenth*(August 2000), 39–48. https://pdfs.semanticscholar.org/f416/161b2f059da3e30eb01921630627a169d552.pdf

4. Mich L., Roberto G., The Natural Language Engineering Approach to Information Systems Development, to be published in Journal of AI*IA (in Italian)

5. Burg J.F.M., Linguistic Instruments in Requirements Engineering, IOS, Amsterdam, 1997

6. Mich, NL-OOPS: From Natural Language to Object Oriented Requirements using the Natural Language Processing System LOLITA, Journal of Natural Language Engineering, Cambridge University Press, 2 (2): 161-187, 1996

7. http://www.pcworld.com/article/219900/IBM_Watson_Wins_Jeopardy_Humans_Rally_Back.html and http://www.forbes.com/sites/bruceupbin/2013/11/14/ ibm-opens-up-watson-as-a-web-service/, accessed 18/12/2014.

8. Ormandjieva O., Hussain I., and Kosseim L. Toward a Text Classification System for the Quality Assessment of Software Requirements Written in Natural Language.

9. Chantree, F., Nuseibeh, B., De Roeck, A., and Willis, A. 2006. Identifying Nocuous Ambiguities in Natural Language Requirements. In Proceedings of 14th IEEE International Requirements Engineering Conference (RE'06), Minneapolis,USA, 59-68.

10. Kim Y, Hurdle J, Meystre SM. Using UMLS lexical resources to disambiguate abbreviations in clinical text. AMIA Annu Symp Proc. 2011;2011:715–22.

11. Lesk M. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. Proceedings of the 5th annual international conference on Systems documentation. 1986: 24-6.

12. Yarowsky D. Word-sense disambiguation using statistical models of Roget's categories trained on large corpora, Proceedings of the 14th conference on Computational linguistics. 1992:23-8.

13. Sussna M. Word sense disambiguation for free-text indexing using a massive semantic network. Proceedings of the second international conference on Information and knowledge management. 1993:67-74.

14. Reyes-Ortiz JA, Gonzalez-Beltran BA, Gallardo-Lopez L. Clinical Decision

Support Systems: A Survey of NLP-Based Approaches from Unstructured Data. Proc - Int Work Database Expert Syst Appl DEXA. 2016;2016-February(September):163–7.

15. D. L. Hunt, R. B. Haynes, S. E. Hanna and K. Smith, Effects of computer based clinical decision support systems on physician performance and patient outcomes: a systematic review, Journal of the American Medical Association, vol. 280, no. 15, pp. 1339-1346, 1998.

16. B.Macias, S.G. Pulman. Natural Language processing for requirement specifications. In Redmill and Anderson, Safety Critical Systems, pages 57-89. Chapman and Hall, 1993.

17. L.Goldin, D.M. Berry. Abstfinder, a prototype Abstraction Finder for Natural Language Text for Use in Requirements Elicitation: Design, Methodology, and Evaluation. First International Conference on Requirements Engineering, 1994.

18. I. Hooks, Writing Good Requirements, Proc. Of the Fourth International Symposium of the NCOSE , 1994, Vol. 2., pp. 197-203.

19. W.M.Wilson, L.H. Rosenberg, L.E. Hyatt. Automated quality analysis of Natural Language Requirement specifications. PNSQC Conference, October 1996.

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

21. E.Kamsties, B.Peach Taming Ambiguity in Natural Language Requirements ICSSEA 2000-5.

22. L. Mich, R. Garigliano, Ambiguity measures in Requirements Engineering, Proc. International Conference on Software - Theory and Practice - ICS2000, 16th IFIP World Computer Congress, Beijing, China, 21-25 August 2000, Feng Y., Notkin D., Gaudel M., Publishing House of Electronics Industry, Beijing, 2000, pp. 39-48.

23. J. Natt och Dag, B. Regnell, P. Carlshamre, M. Andersson, J. Karlsson Evaluating Automated Support for Requirements Similarity Analysis in Market-Driven development Seventh International Workshop on Requirements Engineering: Foundation for Software Quality, Interlaken, Switzerland, June 4-5 2001.

24. Ambriola, V., and Gervasi, V. 1997. Processing natural language requirements. In Proceedings of the 12th international conference on Automated software engineering 36-45.

25. Goldin, L., and Berry, D. M. 1994. Abstfinder, a prototype abstraction finder for natural language text for use in requirements elicitation: design, methodology, and evaluation. In Proceedings of the First International Conference on Requirements Engineering, 18–22.

26. Lee, B. S., and Bryant, B. R. 2004. Automation of software system development using natural language processing and two-level grammar. Radical Innovations of Software and Systems Engineering in the Future. Springer, Heidelberg 219–233.

27. Kamsties, E., Berry, D., and Paech, B. 2001. Detecting ambiguities in requirements documents using inspections. In Proceedings of the First Workshop on Inspection in Software Engineering (WISE'01), 68-80.

28. Fuchs, N. E., and Schwitter, R. 1995. Specifying logic programs in controlled natural language. In Proceedings of the Workshop on Computational Logic for Natural Language Processing, 3–5.

29. Mich, L., and Garigliano, R. 2000. Ambiguity measures in requirement engineering. In Proceedings of international conference on software—theory and practice (ICS2000), 39–48.

30. Kiyavitskaya, N., Zeni, N., Mich, L., and Berry, D. M. 2008. Requirements for tools for ambiguity identification and measurement in natural language requirements specifications. Requirements Engineering Journal 13, 207–240.

31. Agarwal, R., and Boggess, L. 1992. A simple but useful approach to conjunct identification. In Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics, 15–21.

32. Rus, V., Moldovan, D., and Bolohan, O. 2002. Bracketing compound nouns for logic form derivation. In Proceedings of the Fifteenth International Florida Artificial Intelligence Research Society Conference (FLAIRS), 198 – 202.

33. Resnik, P. 1999. Semantic similarity in a taxonomy: An information- based measure and its application to problems of ambiguity in natural language. Journal of Artificial Intelligence Research (JAIR), 11, 95–130.

34. Meyer, B., “On Formalism in Specifications,” IEEE Software, 2(1): pp. 6-26, January 1985.

35. Kamsties, E., Berry, D.M., and Paech, B., “Detecting Ambiguities in Requirements Documents Using Inspections,” p. 68-80 in Proceedings of the First Workshop on Inspection in Software Engineering (WISE'01), Paris, France, July 23, 2001.

36. Letier, E., Kramer, J., Magee, J. and Uchitel, S., "Monitoring and Control in Scenario-Based Requirements Analysis," Proceedings ICSE 2005 - 27th International Conference on Software Engineering, ACM Press, St. Louis, Missouri, USA, May 2005.

37. Ambriola, V., Gervasi, V., "Processing natural language requirements," In proceedings of Automated Software Engineering (ASE'97): 12th IEEE International Conference, November 1-5, 1997, pp. 36-45, 1997.

38. Cyre, W. R., “A Requirements Sublanguage for Automated Analysis,” International Journal of Intelligent Systems, 10 (7), pp. 665-689, July 1995.

39. Denger, C., Berry, D., Kamsties, E., "Higher Quality Requirements Specifications through Natural Language Patterns," SWSTE, p. 80, IEEE International Conference on Software-Science, Technology & Engineering, 2003.

40. Fantechi, A., Gnesi, S., Ristori, G., Carenini, M., Vanocchi, M., and Moreschini, P., “Assisting requirement formalization by means of natural language translation,” Formal Methods in System Design, vol. 4, pp. 243-263, 1994.

41. Rolland, C. and Proix, C., “A Natural Language Approach For Requirements Engineering,” Proceedings of the Fourth International Conference CAiSE'92 on Advanced Information Systems Engineering, vol. 593 of Lecture Notes in Computer Science, pp. 257-277, Manchester, United Kingdom, 1992.

42. Osborne, M. and MacNish, C.K., “Processing natural language software requirement specifications,” In Proceedings of ICRE'96: 2nd IEEE International Conference on Requirements Engineering, pp. 229-236. IEEE Press, 1996.

43. Wilson, W., “Writing Effective Requirements Specifications," USAF Software Technology Conference, Utah, 1997.

44. Wilson, W., Rosenberg, L. and Hyatt, L., “Automated Quality Analysis of Natural Language Requirement Specifications,” 14th Annual Pacific Northwest Software Quality Conference, Portland, 1996.Bowman, B., Debray, S. K., and Peterson, L. L. Reasoning about naming systems. *ACM Trans. Program. Lang. Syst., 15,* 5 (Nov. 1993), 795-825.

45. Fabbrini, F., Fusani, M., Gnesi, S., and Lami, G., “An Automatic Quality Evaluation for Natural Language Requirements,” Proceedings of the Seventh International Workshop on Requirements Engineering: Foundation for Software Quality REFSQ'01, Interlaken, Switzerland, June 4-5, 2001.

46. Lami, G., Gnesi, S., Fabbrini, F., Fusani, M., and Trentanni, G., “An Automatic Tool for the Analysis of Natural Language Requirements,” published as Technical Report 2004-TR-40, Consiglio Nazionale delle Ricerche, Instituto di Scienza e Tecnologie dell'Informazione 'A. Faedo', 2004.

47. Lin, D., “Dependency-based Evaluation of MINIPAR,” In Workshop on the Evaluation of Parsing Systems, Granada, Spain, May, 1998.

48. Sri Fatimah Tjong. 2008. Avoiding ambiguity in requirements specifications.Thesis submitted to the University of Nottingham for the degree of Doctor of Philosophy.

49. Tjong, Sri Fatimah, and Daniel M. Berry. 2013. The Design of SREE—A Prototype Potential Ambiguity Finder for Requirements Specifications and Lessons Learned. Requirements Engineering: Foundation for Software Quality. Springer Berlin Heidelberg, 2013. 80-95.

50. Fuchs, Norbert E., Kaarel Kaljurand, and Tobias Kuhn. 2008. Attempto Controlled English for Knowledge Representation. In Reasoning Web, Springer-Verlag Berlin Heidelberg, LNCS 5224, pp. 104–124.

51. Popescu, D., Rugaber, S., Medvidovic, N., & Berry, D. M. 2008. Reducing ambiguities in requirements specifications via automatically created object-oriented models. In *Innovations for Requirement Analysis. From Stakeholders' Needs to Formal Designs* (pp. 103-124). Springer Berlin Heidelberg.

52. Sven Körner and TorbenBrumm. 2009. RESI-A natural language specification improver. IEEE International Conference on Semantic Computing (ICSC).

53. Kavi Mahesh and Sergei Nirenburg. 1996. Knowledge-based Systems for Natural Language Processing. Chapter Simon Fraser University - Athabasca University, Canada, 1996.

54. Thelin, T., Runeson, P., & Wohlin, C. 2003. An experimental comparison of usage-based and checklist-based reading. Software Engineering, IEEE Transactions on, 29(8), 687-704.

55. Basili, Victor R., Scott Green, Oliver Laitenberger, Filippo Lanubile, Forrest Shull, Sivert Sorumgard. 1995. The Empirical Investigation of Perspective-Based Reading. Technical report the empirical investigation of perspective based reading.

56. Christina Unger and Philipp Cimiano, 2011. Representing and resolving ambiguities in ontology-based question answering.

57. Inah Omoronyia. 2010. A Domain Ontology Building Process for Guiding Requirements Elicitation. In: R. Wieringa and A. Persson (Eds.): REFSQ 2010, LNCS 6182, Springer Verlag Berlin Heidelberg , pp. 188–202.

58. Hassell, Joseph, Boanerges Aleman-Meza, and I. Budak Arpinar. 2006. Ontology-Driven Automatic Entity Disambiguation in Unstructured Text. In : I. Cruz et al. (Eds.): ISWC 2006, LNCS 4273, pp. 44 – 57.

59. Gracia, Jorge, Vanessa Lopez, Mathieu d'Aquin, Marta Sabou, Enrico Motta. 2007. Solving semantic ambiguity to improve semantic web based ontology matching. The 2nd International Workshop on Ontology Matching, Busan, South Korea

60. Anandha Mala and Uma. 2006. Object Oriented Visualization of Natural Language Requirement Specification and NFR Preference Elicitation, In : IJCSNS International Journal of Computer Science and Network Security, VOL.6 No.8, 91-100.

61. Barrett and Beum-Seuk Lee. 2002. Two-level Grammar as an Object-Oriented Requirements Specification Language. Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS'02) -Volume 9 - Volume 9, 280, ISBN:0-7695-1435-9.

62. Tichy, W. F., Landhäußer, M., Körner, S. J., & Fasanengarten, A. nlrpBENCH: A Benchmark for Natural Language Requirements Processing. Manuscript submitted for publication.

63. ‘SBVR version 1.0. January, 2000. http://www.omg.org/spec/SBVR/1.0

64. Bajwa, Imran Sarwar, Behzad Bordbar, and Mark G. Lee. 2010. OCL constraints generation from natural language specification. In Enterprise Distributed Object Computing Conference (EDOC),n2010 14th IEEE International, 2010, pp. 204-21.

65. Al-Harbi, O., Jusoh, S., & Norwawi, N. 2012. Handling Ambiguity Problems of Natural Language Interface for Question Answering.

66. A. Reed and B. Kellogg, Higher Lessons in English., 1877. (In Wikipedia: Sentence Diagram, en.wikipedia.org/wiki/Sentence_Diagram, last accessed: April 21st 2013).

67. Yang, H., Willis, A., De Roeck, A., & Nuseibeh, B. (2010). *Automatic detection of nocuous coordination ambiguities in natural language requirements*. 53. https://doi.org/10.1145/1858996.1859007.

68. Long D., Garigliano R., Reasoning by Analogy and Causality: Model and Applications, Chichester, UK, Ellis Horwood, 1994

69. Morgan R., Garigliano R., Callaghan P., Poria S., Smith M, Urbanowicz A., Collingham R., Costantino M., Cooper C. and the LOLITA Group, Description of the LOLITA System as used in MUC-6, Proc. 6th ARPA Message Understanding Conf., Morgan Kaufmann, 1996

70. Garigliano R., Boguraev B., Tait J., Editorial, Journal of Natural Language Engineering, Cambridge University Press 1 (1): 1-7, 1995

71. Robin. 2009. POS-Tagging. Available at http://language.worldofcomputing.net/pos-tagging, Dec.4th 2009.

72. Tagger.Available at http://nltk.googlecode.com/svn/t runk/doc/ h owto/tag.html

73. Aronson AR. Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. Proc AMIA Symp 2001:17-21.

74. Fan R, Chang K, Hsieh C, Wang X, Lin C. LIBLINEAR: A library for large linear classification. Journal of Machine Learning Research. 2008;9:1871-4. Software available at http://www.csie.ntu.edu.tw/~cjlin/liblinear

75. Alias-i. LingPipe. 2010; Available from: http://alias-i.com/lingpipe/.

76. NLM. UMLS SPECIALIST Lexicon. 2010; Available from: http://lexsrv3.nlm.nih.gov/SPECIALIST/index.html.

77. Savova GK, Masanz JJ, Ogren PV, Zheng J, Sohn S, Kipper-Schuler KC, et al. Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications. J Am Med Inform Assoc. 2010 Sep-Oct;17(5):507-13.

78. Pakhomov S. Semi-Supervised Maximum Entropy Based Approach to Acronym and Abbreviation Normalization in Medical Texts. Proceedings of 40th Annual Meeting of the ACL. 2002:160-7.

79. E. A. Mendona, J. Haas, L. Shagina, E. Larson and C. Friedman, Extracting information on pneumonia in infants using natural language processing of radiology reports, Journal of biomedical informatics, vol. 38, no. 4, pp. 314-321, 2005.

80. S. Meystre and P. J. Haug, Natural language processing to extract medical problems from electronic clinical documents: performance evaluation, Journal of biomedical informatics, vol. 39, no. 6, pp. 589-599, 2006.

81. B. Hazlehurst, H. R. Frost, D. F. Sittig and V. J. Stevens, MediClass: A system for detecting and classifying encounter-based clinical events in any electronic medical record, Journal of the American Medical Informatics Association, vol. 12, no. 5, pp. 517-529, 2005.

82. F. J. Friedlin and C. J. McDonald, A software tool for removing patient identifying information from clinical documents, Journal of the American Medical Informatics Association, vol. 15, no. 5, pp. 601-610, 2008.

83. K. B. Wagholikar, K. L. MacLaughlin, M. R. Henry, R. A. Greenes, R. A. Hankey, H. Liu and R. Chaudhry, Clinical decision support with automated text

processing for cervical cancer screening, Journal of the American Medical Informatics Association, vol. 19, no. 5, pp. 833-839, 2012.

84. M. Fiszman, W. W. Chapman, D. Aronsky, R. S. Evans and P. J. Haug, Automatic detection of acute bacterial pneumonia from chest Xray reports, Journal of the American Medical Informatics Association, vol. 7, no. 6, pp. 593-604, 2000.

85. N. L. Jain, C. A. Knirsch, C. Friedman and G. Hripcsak, Identification of suspected tuberculosis patients based on natural language processing of chest radiograph reports, in Proceedings of the AMIA Annual Fall Symposium, American Medical Informatics Association, Washington, USA, pp. 542546, 1996.

86. A. Roberts, R. Gaizauskas and M. Hepple, Extracting clinical relationships from patient narratives, in Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing, Association for Computational Linguistics, Ohio, USA, pp. 10-18, 2008.

87. A. J. Watson, J. O'Rourke, K. Jethwani, A. Cami, T. A. Stern, J. C. Kvedar and A. H. Zai, Linking electronic health record-extracted psychosocial data in real-time to risk of readmission for heart failure, Psychosomatics, vol. 52, no. 4, pp. 319-327, 2011.

88. M. E. Matheny, F. FitzHenry, T. Speroff, J. K. Green, M. L. Griffith, E. E. Vasilevskis and S. H. Brown, Detection of infectious symptoms from VA emergency department and primary care clinical documentation, International journal of medical informatics, vol. 81, no. 3, pp. 143-156, 2012.

89. R. J. Byrd, S. R. Steinhubl, J. Sun, S. Ebadollahi and W. F. Stewart, Automatic identification of heart failure diagnostic criteria, using text analysis of clinical notes from electronic health records, International journal of medical informatics, vol. 83, no. 12, pp. 983-992, 2014.

90. K. W. Fung, C. S. Jao and D. Demner-Fushman, Extracting drug indication information from structured product labels using natural language processing, Journal of the American Medical Informatics Association, vol. 20, no. 3, pp. 482-488, 2013.

91. T. D. Imler, J. Morea, C. Kahi and T. F. Imperiale, Natural language processing accurately categorizes findings from colonoscopy and pathology reports, Clinical Gastroenterology and Hepatology, vol. 11, no. 6, pp. 689-694, 2013.

92. Shah, U. S., & Jinwala, D. C. (2015). Resolving Ambiguities in Natural Language Software Requirements. *ACM SIGSOFT Software Engineering Notes*, *40*(5), 1–7. https://doi.org/10.1145/2815021.2815032

93. C. Denger, J. D€orr, and E. Kamsties, QUASAR: A survey on approaches for writing precise natural language requirements. [Online]. Available: http://publica.fraunhofer.de/eprints/urn: nbn:de:0011-n-77930.pdf, 2001.

94. S. Withall, Software Requirement Patterns (Best Practices), 1st ed, Redmond, WA, USA: Microsoft, 2007.

95. K. Pohl and C. Rupp, Requirements Engineering Fundamentals, 1st ed, Rocky Nook, Santa Barbara, CA 93103, 2011.

96. E. Uusitalo, M. Raatikainen, T. Mannisto, and T. Tommila, "Structured natural language requirements in nuclear energy domain towards improving regulatory guidelines," in Proc. 4^{th} Int. Workshop Requirements Eng. Law, pp. 67–73, 2011.

97. K. Pohl, Requirements Engineering-Fundamentals, Principles, and Techniques, New York, NY, USA: Springer, 2010.

98. D. Berry, E. Kamsties, and 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

99. Arora C, Sabetzadeh M, Briand L, Zimmer F. Automated checking of conformance to requirements templates using natural language processing. IEEE Trans Softw Eng. 2015;41(10):944–68.

100. A. Mavin, P. Wilkinson, A. Harwood, and M. Novak, "Easy approach to requirements syntax (EARS)," in Proc. 17th IEEE Int. Requirements Eng. Conf., 2009, pp. 317–322.

101. A. Mavin and P. Wilkinson, "Big EARS (the return of "Easy Approach to Requirements Engineering")," in Proc. 18th IEEE Int. Requirements Eng. Conf., 2010, pp. 277–282.

102. S. Gregory, "Easy EARS: Rapid application of the easy approach to requirements syntax," in Proc. 19th IEEE Int. Requirements Eng. Conf., 2011, pp. 1–2.

103. B.Welch Practical Programming in Tcl and Tk second edition Prentice Hall 1997.

104. Levenshtein VI. Binary codes capable of correcting deletions, insertions, and reversals. Soviet Physics Doklady. 1966;10(8): 707-10.

105. Yarowsky D. Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. ACL-95. Cambridge, MA: ACL; 1995:189-96.

106. Tichy, W. F., Landhäußer, M., & Körner, S. J. (2015). nlrpBENCH: A Benchmark for Natural Language Requirements Processing. *Multikonferenz Software Engineering & Management 2015*, 159–164.

107. Fabbrini, F., Fusani, M., Gervasi, V., Gnesi, S., & Ruggieri, S. (1998). Achieving Quality in Natural Language Requirements. *Proceedings of the 11th International Software Quality Week, February*, 4–5.

108. Fatwanto, A. (2013). Software requirements specification analysis using natural language processing technique. *2013 International Conference on Quality in Research, QiR 2013 - In Conjunction with ICCS 2013: The 2nd International Conference on Civic Space*, 105–110. https://doi.org/10.1109/QiR.2013.6632546

109. Ormandjieva, O., Hussain, I., & Kosseim, L. (2007). Toward a text classification system for the quality assessment of software requirements written in natural language. *SOQUA'07: Fourth International Workshop on Software Quality Assurance - In Conjunction with the 6th ESEC/FSE Joint Meeting*, 39–45. https://doi.org/10.1145/1295074.1295082

110. Bas Basten and Tijs van der Storm. 2010. AMBIDEXTER: Practical Ambiguity Detection Tool Demonstration. Source Code Analysis and Manipulation (SCAM),

2010 10th IEEE Working Conference, 101-102.

111. Liu H, Lussier YA, Friedman C. A study of abbreviations in the UMLS. Proc AMIA Symp. 2001:393-7.