A Bird’s Eye View of Natural Language Processing and Requirements Engineering

Assad Alzayed[1], Ahmed Al-Hunaiyyan[2]
Computer Science and Information Systems Department, College of Business Studies
Public Authority for Applied Education and Training, (PAAET), Kuwait

***Abstract*—Natural Language Processing (NLP) has demonstrated effectiveness in many application domains. NLP can assist software engineering by automating various activities. This paper examines the interaction between software requirements engineering (RE) and NLP. We reviewed the current literature to evaluate how NLP supports RE and to examine research developments. This literature review indicates that NLP is being employed in all the phases of the RE domain. This paper focuses on the phases of elicitation and the analysis of requirements. RE communication issues are primarily associated with the elicitation and analysis phases of the requirements. These issues include ambiguity, inconsistency, and incompleteness. Many of these problems stem from a lack of participation by the stakeholders in both phases. Thus, we address the application of NLP during the process of requirements elicitation and analysis. We discuss the limitations of NLP in these two phases. Potential future directions for the domain are examined. This paper asserts that human involvement with knowledge about the domain and the specific project is still needed in the RE process despite progress in the development of NLP systems.**

Keywords—Automated text understanding; natural language processing; requirements engineering; requirements elicitation

I. Introduction

Natural Language Processing (NLP) has a significant functional value in many application fields. NLP is especially useful in the requirements engineering (RE) domain. RE is a vital part of software engineering and considered as the first phase in the software development life cycle. It consists of several activities, including elicitation, analysis, documentation, validation, and management of requirements [1]. RE is a complicated process that is both time-consuming and error-prone, especially for large projects [2]. The RE process defines all of the requirements that a new system needs to complete to be successful. In addition, RE process collects the necessary and appropriate domain knowledge that comprises the requirements of the stakeholders (customers, end-users, domain experts). The requirements elicitation and requirements analysis tasks are performed incrementally and iteratively to accomplish this goal. These tasks use both informal natural language (NL) and formal modeling language. Using NL helps communicate with the stakeholders. It is the universal format language understood by end-users and stakeholders from all involved disciplines [3]. However, NL can be ambiguous and result in misunderstandings concerning the definition of requirements.

NLP can improve the communication between requirement engineers and domain experts (i.e., the users) by creating suitable RE specification documents [3]. NLP can also improve computer understanding of natural language text written by humans [4].

This paper presents a survey of how NLP supports current RE approaches. Specifically, the following research questions are addressed:

Q1: What is the current state of the practice for elicitation and analysis phases in RE using NLP as the platform?

Q2: What are the activities for the requirements elicitation and analysis in RE using NLP as the platform?

Q3: Are NLP systems improving the requirements elicitation and analysis for RE?

Q4: What are the current limitations of using NLP in requirements elicitation and analysis?

A literature review was conducted to address these research questions. This literature review summarized the research reporting use of NLP to support RE activities. The literature review was divided into data preparation, data collection, and data analysis stages. First, the literature search criteria were developed based on the research questions. Second, literature searches were conducted over a predefined collection of databases, including Springer Link, Scopus, IEEEXplore, Google Scholar, Science Direct, and ACM Digital Library. The literature search results were evaluated by title and abstract for all four study questions. This literature review was not an exhaustive review. Instead, this literature review provides a snapshot of the state-of-the-art practices based on the Kitchenham guidelines [5] for conducting systematic literature reviews.

This paper provides an overview of the current state of practices and challenges associated with the elicitation and analysis phases of RE employing NLP techniques. This study focused on three important aspects:

- Providing an overview of the challenges facing the requirements elicitation and the requirements analysis (i.e., ambiguity, inconsistency, incompleteness, and requirements classification).

- Providing an overview of the state-of-the-art approaches to NLP support of current RE operations. Precisely, this study focused on how the available NLP tools and techniques support requirements elicitation and requirements analysis.
- Providing an overview of the limitations of NLP use in requirements elicitation and requirements analysis.

This paper is organized as follows. Sections II and III provide background discussion about NLP and RE. Section IV provides an overview of RE activities where NLP was used to support the requirements elicitation and analysis processes. Section V discusses current requirements elicitation and analysis practices in NLP. Section VI discusses the current limitations of using NLP in requirements elicitation and analysis. Section VII provides general discussions. Finally, Section VIII concludes the paper.

II. Requirements Engineering

The elicitation, analysis, and management of requirements on the basis of semantics during the system development process are difficult due to a large number of requirements for large system engineering projects. Experts normally face numerous constraints during the processes of elicitation and analysis. These constraints include time restraints, insufficient human cognitive capacity to understand the full scope of the processes, and the volume of data to be processed [6]. These considerations make it difficult to manage and maintain the quality of the software requirements specification (SRS) document.

Requirements elicitation involves understanding the objectives and motivation for proposed system software. This phase usually begins with fundamentally informal knowledge and involves unfamiliar people with the processes for developing a software system. Thus, data interpretation is influenced by misunderstandings between the analysts and the consumers [7]. In RE, it is essential to have an excellent semantic understanding of the situation before beginning the process of requirements elicitation [8]. Traditionally, requirements were communicated to the elicitation team using NL text to prevent misunderstandings related to variations in terminology. However, ambiguity is a problem related to NL [9]. The elicitation team may misinterpret or misunderstand the stated requirements specified using NL due to the method used to communicate these requirements. Additionally, problems can occur that are associated with the elicitation of functional requirements (FRs) and the numerous sub-categories of the non-functional requirements (NFRs) [10]. These problems stem from differences in computer jargon and terminology to describe the requirements between the stakeholders and the requirements engineers [11]. The lack of consistency in the requirements documentation process makes the requirements classification process difficult and prone to error [12]. Cordes and Carver (1989) [13] stated that requirements are not created by a single human individual but are the result of common needs from multiple communities. These requirements can introduce uncertainty and inconsistencies. They further state that different participants in the requirements elicitation process have different interpretations of the meaning of the requirements. The participants have different opinions about the design of the new system based on their interpretation of the requirements. The resulting requirements will be ambiguous, contradictory, and incomplete if the participants in an elicitation process do not have a common semantic understanding of the requirements.

Requirements analysis involves understanding and assessing the documented requirements. This phase is concerned with checking the set of elicited requirements for qualifying conflict, omission, duplication, ambiguity, and inconsistency criteria. Common requirements analysis practices involve using checklists for analysis, prioritization, and sorting of requirements, using interaction matrixes to define differences and overlaps, and developing a risk evaluation of the requirements [14].

Requirements elicitation and requirements analysis are usually interlinked processes. Requirements are identified during elicitation, and then analyses are performed. If issues are identified, they are addressed and solved using the source of the requirements [14]. Once a requirement has been elicited, modeled, and analyzed, the SRS document should be clearly and unambiguously documented [15]. The SRS is part of a contract and must simply, accurately, and unambiguously define the requirements of the user and the system. An SRS that is inconsistent, unmanaged, vague, incorrect, or unclear inevitably leads to cost overruns and missed deadlines [16], [17] and [18]. A noteworthy research issue in RE is ambiguity which is described as “a statement having more than one meaning.” Ambiguities may be lexical, syntactic, semantic, pragmatic, vague, generic, or linguistic [19].

The use of NLP in RE is critical because requirement specifications are developed in collaboration between the software analysts and the end users. End users, consumers, and customers will not sign a contract if the requirements are written in the formal language. [20].

Several projects have demonstrated that the RE process can be automated or semi-automated by using NLP [21], [22], and [23]. Furthermore, NLP can support requirements elicitation and requirements analysis by automatically eliminating the ambiguity barrier.

III. Natural Language Processing

NLP is technically one of the sub-fields of artificial intelligence. NLP implements computational and linguistics techniques to assist computer understanding. Additionally, NLP can create human language in the form of texts and speech/voice [24]. The processing of NL is difficult and involves different techniques from those used in artificial languages [25]. NLP approaches are usually based on machine learning (ML). For NLP, the ML process is composed of two tasks: natural language understanding (which is the task of understanding the text) and natural language generation (NLG) (which is the task of generating text with a syntax that is widely used by humans) [26] and [27]. Another study [28] identified three types of NLP technologies: NLP techniques, NLP tools, and NLP resources. An NLP technique is a functional method, approach, process, or procedure for conducting a specific NLP task. NLP techniques include part-of-speech (PoS) tagging, parsing, and tokenization. An NLP tool is a software system or software library that continues to support one or more NLP techniques. Examples of NLP tools include Stanford CoreNLP7, NLTK8, and OpenNLP9. An NLP resource is a linguistic data resource that assists NLP techniques or tools. An NLP resource can be a language lexicon (i.e., dictionary) or a corpus (i.e., a collection of texts).

A lexical analysis can be included in a requirements document to specify pre-built dictionaries, databases, and rules. The goal of lexical analysis is to analyze the meaning of specific words. Five key techniques can be used in a lexical analysis: sentence splitting, PoS tagging, tokenization, morphological analysis, and parsing. Sentence splitting involves separating the text into different sentences. During this process, the NL text is evaluated to determine sentence boundaries.

Tokenization involves dividing a sentence into meaningful components, called tokens. Depending on the form of the text, which is partially determined by the sentence splitting, the tokens are assigned to a category, including punctuation marks, numbers, symbols, and words. The PoS tagging method involves tagging each token with its grammatical group, depending on its meaning and context. Each token is designated with a tag, including noun, verb, adjective, or determinant [29].

The morphological analysis is the initial stage in syntactic analysis. The goal of syntactic analysis is to define the origin of a compound word. Compound words are quickly stemmed and lemmatized. Stemming is the process that reduces a compound word into its ground form or origin. Lemmatization is the process that searches for the ground form of a word [29].

Parsing is a process that analyzes a sentence by taking each word and determining its structure based on its constituent sections. Two components are required to parse a piece of text: a parser and a grammar. Ambiguous sentences may require several different types of analyses in the grammar of NLs [30]. There are two main parsing approaches: dependency parsing and phrase structure parsing. Dependency parsing focuses on the connections between the words in a sentence. Phrase structure parsing involves construction of a parse tree using probabilistic context-free grammar.

The output of a lexical analysis serves as the input for a syntactic analysis. This method performs a sentence analysis of the words to determine the grammatical structure of the sentence. It requires both grammar and a parser. This level of processing results in a representation of the sentence that shows the structural relationship of the dependence between words [4].

Semantic processing defines the potential meaning of a sentence by focusing on associations between word-level meanings in a sentence [4]. Semantic processing builds a description of the objects and actions identified in a sentence and include the details given by the adjectives, adverbs, and prepositions [31].

The goal of categorization is to automatically assign new documents to categories that are already defined [32]. In RE, the NLP method is used to collect requirements from a text; analyze the consistency, linkages, similarities, and ambiguities in the text; and automatically group the text. It also classifies requirements for specific purposes that may be useful during software development. Work associated with the classified requirements may be split between different software development teams, with each team assigned a different class of requirements [33].

IV. NLP for Requirements Elicitation and Analysis

Traditionally, requirements elicitation and requirements analysis are manually processed, expensive, time-consuming, and resource-demanding [17]. Researchers have applied NLP techniques and tools to support a range of linguistic analysis tasks performed at various phases to produce complete requirements documents without ambiguity and inconsistency [34]. The requirements can be illustrated for the stakeholders in a semi-automated or automated way [35], [36], and [37]. Requirements may appear in different forms, including lists of single words, phrases, paragraphs, short texts, and special characters.

Generally, requirement engineering problems are primarily caused by heavy dependence on humans use of NL [38]. NL is syntactically ambiguous and semantically inconsistent. A systematic analysis of literature from 1995 to 2016 indicates that assembly of ambiguous requirements remains one of the most critical problems in software engineering [39]. In response, researchers have attempted to use NLP systems to solve the ambiguity challenges of NL. NLP systems have also been used to support the communication process between system users and stakeholders during the development stages of a system [15]. Communication techniques may focus on pre-selected tools (e.g., Stanford Parser), preferred methods (e.g., rule-based and ontological-based), or degree of automation. The work of [40] provides a detailed discussion about the current approaches to ambiguity in the field of requirements. This paper evaluates empirical work on NLP tools and techniques for dealing with different types of requirement ambiguity [40]. These studies indicate that a significant number of current software implementations solely rely on ambiguity recognition. However, compensation is the responsibility of the stakeholders.

An interesting research area is using NLP for eliciting and analyzing domain requirements based on developed domain ontologies. Ontologies provide a standardized means of organizing information among stakeholders in RE. Thus, ontologies may significantly enhance the quality of the elicited requirements [41]. For example, [42] used a domain ontology and meta-model requirements to generate and elicit requirements. Similarly, [43] describes three core features of domain ontologies ideal for elicitation requirements: explicit relational expression, competent relationship recognition, and explicit temporal and spatial expressions. For the creation of certain domain ontologies, A rule-based approach is recommended for the creation of certain domain ontologies from NL technical documents [44]. The research [45] used NLP to derive formal representations of the requirements based on object-oriented designs using intermediate models.

NLP is recognized as general assistance in analyzing requirements for ambiguity defection [46]. NLP techniques were used to retrieve information and synthesize models. For example, [47] produced unified modeling language (UML) templates (e.g., use-case, analysis class, collaboration, and design class diagrams) from the requirements of natural language using a collection of syntactic reconstruction rules. In addition, [48] proposed a tool-supported approach to promote the process of requirements analysis and the retrieval of class diagrams from textual requirements that support NLP and domain ontology techniques.

Emerging software paradigms, including social networks, mobile computing, and cloud computing, has expressed a growing interest in using NLP techniques. Additionally, NLP techniques are being explored for extensive data analysis to enable data-driven RE [49] and crowd-based RE [50]. Requirements articulated in user stories have been presented as an interesting application of NLP to support agile methodology [51].

V. Current State of Practice of NLP in RE

Applying NLP to RE is an area of research and development that implements NLP tools, techniques, and resources to a range of requirements documentation to facilitate various linguistic analysis activities performed at different RE phases. These tasks include detecting language problems, defining core domain terms, and creating traceability links between requirements [28].

Currently, most NLP tools are used for solving problems in the elicitation phase. NLP tools are also used to extract NL text by the process model, based on parsers and tagging [52]. For example, [53] used 2PoS tagging for preprocessing during the development of conceptual models. This paper proposed an automated solution called Visual Narrator based on NLP. Visual Narrator derives a conceptual model based on user story requirements. In this process, the PoS tagging is used to define the linguistic pattern of the sentence. If the PoS tagging is determined to be a requirement, it is collected from the text corpus and gathered for the next steps in the methodology. The automated approach enables identifying dependencies, redundancies, and inconsistencies between requirements based on a comprehensive and understandable view created from long textual requirements.

The latest trend in requirements elicitation uses NLP is to mine accessible databases (e.g., social media, requirement documents, or Apple Store feedback). The mining process is carried out with the help of ML techniques, NLP, and text mining [54]. Recently, a growing body of research has assessed the use of NLP techniques to extract requirements based on different types of user feedback for the requirements elicitation process [55]. For example, [56] created a tool for detecting ambiguous words in translated SRS.

Currently, an approach has been used to automate requirements elicitation and classification criteria which utilizes an intelligent conversational chatbot. For example, [57] used ML and artificial intelligence to develop a chatbot that interacts with stakeholders using NL and creates formal system requirements based on conversation. This chatbot then classifies the elicited requirements into functional and non-functional system requirements. Additionally, chatbots are widely used in web applications to provide help or information requested by users. For example, CORDULA is a framework that uses chatbot technology to establish contact with end-users for requirements elicitation and understand users' needs. CORDULA guides the users to their desired outcome with minimal effort required by the end-user [58].

Domain ontology has been widely used to improve the elicitation and analysis of functional and NFRs. For example, [59] used NLP to extract NFRs for natural language documents. Furthermore, [60] used an ontology-based approach to support the collection of knowledge to identify possible solutions for eliciting NFRs. Additionally, NLP has been used in similarity analyses to identify functional and NFRs from user app reviews [61] and [62]. The study of [63] proposed an ontology-based approach to support software requirements traceability, which makes it possible for a development team to effectively manage the evolution of the requirements for a software product.

New requirements analysis tools based on NLP are emerging. These tools should significantly reduce the cost of fixing requirement errors by faster identification, thus freeing domain experts from tedious, time-consuming tasks. For example, QuARS (Quality Analyzer for Requirements Specifications) is a tool that analyzes NL requirements in a comprehensive and automated manner using NLP techniques. QuARS emphasizes detecting potential linguistic weaknesses (i.e., ambiguity) that can create issues with interpretation at the next stage of software development [64]. This tool partially assists with the analysis of accuracy and completeness by grouping requirements based on specific concerns. However, user interaction is recognized as a crucial factor adversely affecting the performance and approval of the entire processing. The study of [65] proposed an NLP technique that uses a classification method to automatically handle redundancy and inconsistency problems in a requirement document.

An annual workshop called the NLP4RE (Natural Language Processing for Requirements Engineering) was established to explore interests in NLP applications related to RE issues [66]. The goal of NLP4RE is to help requirements analysts perform multiple linguistic analysis activities for RE phases. This workshop has produced numerous publications and gained broad interest from diverse cultures. About 42.7% of NLP4RE studies focused on the analysis phase. These analysis phase studies used detection as the core linguistic analysis activity and requirements specifications as the processed document type [28].

In the industrial sector, many companies have begun developing NLP tools for RE. For example, Qualicen developed Requirements Scout, a tool that analyses requirement specifications, detects requirement ambiguity, and requirement “smells.” The ThingsThinking system offers several tools under the brand name, Semantha®. This system includes a tool that classifies requirements and identifies the associated risks. It also has a tool that performs document comparison on a semantic level, which can be used for analysis of requirements documents created by multiple stakeholders. QRA Corp developed QVscribe, a tool that checks the quality and consistency of requirements documents. OSSENO Software developed ReqSuite, which is a tool to support the writing and review of specifications. IBM recently developed the IBM Engineering Requirements Quality Assistant, which is an application that leverages the advanced NLP capabilities of IBM Watson for automated requirements analysis and management.

VI. Current Limitations of NLP in Requirements Elicitation and Requirements Analysis

An automated means of enabling software engineers and project managers to develop and refine their NL requirements is needed in RE. NLP can reduce the human effort in making NL requirements clear, consistent, unambiguous, and easy to understand by all stakeholders before moving into modeling and design phases. There are still numerous limitations on the capabilities and rationales for using NLP techniques, despite research developments on NLP for RE. Various challenges for NLP use within requirements elicitation and analysis still exist, including the followings:

- Coreference resolution: Coreference resolution is the task of extracting several expressions in a sentence or text that refer to the same entity/actor in a requirements document. It is especially employed at the semantic/pragmatic level when two nouns are treated the same [67]. Coreference resolution is a key challenge of NLP, not only in English but also for all other languages [68].

- Emotion Detection (ED): While the use of NLP system in requirements elicitation and feedback techniques are well defined, there are no current state-of-the-art techniques that combine emotionally driven features and the capture of user feedback on these features [69]. Emotion recognition may also be used to evaluate social media data or to spot fake news [70]. A major challenge in ED is that the cultural affiliations of an individual may significantly impact their expressed feelings in a situation [69]. However, progress is being made as several methods have been developed to solve this problem. These methods include the use of a knowledge-enriched transformer [71], focusing on latent representation [72], and building new datasets that focus on emotions [73] and [74].

- Unimodal LNP: Current NLP systems are primarily unimodal. Thus, they are limited to process and analysis of linguistic inputs [75]. However, humans are multimodal. They use diverse combinations of visual, auditory, tactile, and other inputs. Humans do not handle each sensory model in isolation but rather simultaneously. This process incorporates each sensory model to enhance the quality of awareness and understanding [76]. Therefore, from a computational point of view, NLP needs to have these same abilities to achieve human-level ground and understanding in a variety of AI tasks. NLP must be assisted by multimodal control interfaces, identification and understanding of human behavior, and collaborative decision-making between the system and individuals or groups to understand the requirements of the customer and other stakeholders [77]. Visual question answering is a method that addresses the challenging unimodal aspect of NLP systems [78]. Many other methods are used to integrate multimodality into NLP structures, including declarative learning-based programming [79], multimodal datasets [80], procedural reasoning networks [81], and unified attention networks [82].

- Ability to recognize requirement sentences that contain contextual information rather than merely describing the process steps [28]. The inherent ambiguity of NL can lead to differing interpretations of the same sentence [83].

- Domain ontologies approach: [43] found that requirement analysts were more likely to misidentify concepts and relationships when using a domain ontologies approach. Thus, domain ontologies need to be investigated to develop a deeper understanding of the requirements and their respective relationships [84].

- NLP accuracy in extracting the correct requirements must be improved. NLP must be enhanced by other methods (e.g., ML) to eliminate errors. Accuracy must be substantially improved if NLP to be seriously considered for use with RE [85].

- The algorithms for detecting ambiguity need improvement and fine-tuning while simultaneously avoiding over-fitting. These improvements are needed to evaluate whether the use of domain ontologies can lead to a deeper understanding of the requirements and their relationships [86].

- PoS detection is generally considered a challenge that has been resolved. However, there are still issues with incorrect POS tags [87].

Attempting to resolve all ambiguities in a requirements specification is a time-consuming process that cannot be fully automated. Human interaction is needed to overcome dynamic ambiguities that are dependent on domain knowledge. Controlled language is helpful in identifying or avoiding ambiguities in SRS. However, the input must be written in the constraint language, and lexical and syntactic ambiguities must be addressed. Furthermore, methods that use knowledge-based, ML, and ontology techniques may produce precise outcomes by detecting semantic ambiguities in the requirements specifications [88], [89], [90].

VII. Discussion

Our research focused on a specific field and evaluated a range of trends explained and summarized in this section. The objective of this research was to provide a state-of-the-art summary of NLP performed in various RE activities. This research is intended to be an overview for domain experts and serve as an entry point for researchers in this area. We present results based on the conducted literature review. As previously discussed, the literature review was not entirely systematic.

Thus, our findings may be revised and/or expanded by future studies within this domain.

Table I and Section IV provide answers to RQ1 ("What is the current state of the practice for elicitation and analysis phases in RE using NLP as the platform?") and RQ2 ("What are the activities for the requirements elicitation and analysis in RE using NLP as the platform?"). Twenty-five articles (i.e., the "Contributions" column in Table I) were closely analyzed to assess the state-of-the-art of NLP the use NLP in RE activities. We assume that most studies are preliminary proposals because there are more academic research papers than actual software projects for industrial applications. As listed in Table I, the "NLP Tasks" and "RE Support.

TABLE I. CONTRIBUTIONS AND RE ACTIVITIES SUPPORTED BY NLP

N	Contribution	Publication Year	Objective/Purpose	NLP Tasks	RE Activities Support	Tools +Techniques + Resources	User interaction
1	Shah (+)	2015	Compare NLP approaches to resolve ambiguity	Detection	Elicitation + Analysis	-	-
2	Omoronyia et al., (-)	2010	Reducing the effort of building a domain ontology for requirements elicitation by analyzing NL text from technical standard	Extracting semantic graphs	Elicitation	POS tags Sentence parser	Medium
3	Ibrahim et al., (*)	2010	Using NLP and domain ontology to extract UML class diagram from informal NLP by using the RACE tool to produce class diagram from requirements	Extracting	Analysis	PoS tags, Open NLP parser	Medium
4	Maalej et al., (*)	2015	The authors discussed how software development organizations can employ user feedback to identify, prioritize, and manage requirements	Classification	Elicitation + Prioritization	-	-
5	Bano (+)	2015	The findings of a mapping analysis are discussed in a collection of empirical work from the last two decades that discuss the concept of complexity in RE using NLP methods and techniques	Ambiguity Detection	-	-	-
6	Robeer et al., (+)	2016	Extracting conceptual modules from user stories requirements using NLP	Modeling	Analysis	spaCy tagger	Low
7	Bordignon et al. (-)	2018	Using NLP PoS tagging and Parsing in the first three BPM phases to extract ambiguity	Classification	Elicitation + Analysis	PoS Tagging	Low
8	Lucassen et al., (-)	2017	Proposing an automated approach called Visual Narrator that is based on extraction of conceptual models from user story requirements	Classification + Clustering	Elicitation +Analysis	PoS tagging in order to further optimize the results	Low
9	Groen et al., (*)	2015	Proposes crowd-based RE that integrates elicitation and analysis techniques using a crowd sourcing concept where individual tasks are aggregated to provide a final list of identified requirements	Clustering -	Elicitation		
10	Cruz et al. (-)	2017	Proposes two approaches to identify vague words and phrases in the requirements document in a multilingual language	Classification	Elicitation	-	Low
11	Friesen et al. (*)	2018	Developed a chatpot to help end-users revise current extraction and classification requirements	Detection	Elicitation	POS Tagging	High
12	Hollis et, al., (*)	2017	Proposes a method to capture verbal discussion and translate it into a text transcript, a real time audio-to-text conversion software		Elicitation and Specification	Text mining	
13	Murtazina and Avdeenko (*)	2019	Proposes a method to convert knowledge domain into ontology language	Classification	Traceability	-	Medium
14	Zhong, et, al., (+)	2019	Proposes a Knowledge Transformer (KET) to detect emotions in a conversation	Detection + Classification	Sentiment analysis	-	-
15	Ernst and J. Mylopoulos (-)	2010	Eliciting FR and NFR requirements	Classification	Elicitation	-	-

16	Hu, et, al., (*)	2016	Investigates the lack of consistency in RE and the feasibility of using Human Error Taxonomy to support the SRS inspection process	Detection + Classification	Analysis	-	Low
17	Gnesi et, al., (*)	2019	Using NLP techniques with an emphasis on detection of potential linguistic weaknesses (ambiguity)	Classification Clustering	Analysis	Syntax Parser	High
18	Acheampong, at, al., (+)	2020	A survey of state of the art NLP techniques that combines both emotionally-driven features and the capture of user feedback on these features	Sentiment	Analysis	-	-
19	Abdul-Mageed, et, al., (-)	2017	Using Twitter information to create a large dataset of fine-grained emotions for deep learning purposes	Classification	Analysis	-	-
20	Kordjamshidi, et, al., (+)	2017	Integrated multimodality into NLP structures, including declarative learning-based programming	Classification	Analysis	-	Low
21	Van der Aa et, al., (-)	2018	Describes an application of NLP in the BPM context and illustrates how NLP has the ability to maximize the advantages of BPM activities at multiple stages	-	-	-	-
22	Dalpiaz, et, al., (-)	2018	Proposes a visual model to detect ambiguity in requirements that maybe triggered by the use of different words to refer to the same concept	Classification	Analysis	Semantic Similarity Algorithms	Low
23	Femmer, et, al., (-)	2018	NLP toll to detect quality findings and checks on SRS smells in the requirements	Classification	Analysis	is Word- and Sentence Splitting, Morphologic analysis, Lemmatization (and sometimes stemming), PoS tagging, and syntactic parsing	Low
24	Zhao, et, al., (-)	2020	Presents a comprehensive overview of the applications of NLP in RE	-	Elicitation Analysis	-	-
25	Mezghani, et, al., (-)	2018	Proposes an approach using NLP to detect technical business terms associated with the requirements documents	Classification + Clustering	Analysis	PoS tagging + Noun chunking	

Legends: (+) improves state of the art; (*) no information related to state of the art; (-) comparable with state of the art

Activities” columns address RQ2 and describe the various RE tasks that can be assisted by NLP techniques. These RE tasks include traceability, ambiguity detection, and requirements classification. The available techniques and tools developed to support each RE task are presented (e.g., PoS, tagging, and tokenization). The “Contributions” column in Table I also includes a partial response to RQ 3 (“Are NLP systems improving the requirements elicitation and analysis for RE?”). The answer to RQ3 appears to be preliminary; as indicated by the lack of comparison with state of the art in Table I. Table I also include information related to the quantity the NLP data and user expectations, emotions, and experiences. This rich data set may be used by software developers to assess better their product users' needs, experiences, and sentiments. Mining NLP, especially user opinions, can yield valuable information for product upgrades by software development organizations. However, it is often difficult to extract user requirements from massive amounts of data. Opinions are often shared without regard to grammar or style. This issue has recently caused problems with corpus processing. As a result, we assume that the collected data are unstructured. Software developers focus on user feedback for requirements elicitation and analysis. However, the trustworthiness of comments, tweets, or feedback remains a major problem for the software development community. In section VI, we discussed a variety of limitations in this domain. Other challenges or limitations that the RE community faces by using NLP as the source for eliciting user requirements include user privacy and personalization [55]. These findings address RQ4 (“What are the current limitations of using NLP in requirements elicitation and analysis?”).

Table I lists the NLP methods used in the articles reviewed in this paper. Most researchers use NLP techniques for the identification of ambiguity in RE. Furthermore, due to its wide role in RE, the analysis process was the phase with the most attention in the research. We observed that NLP was primarily used in the preprocessing phase to convert data into a format that was consumable by all stakeholders. Most of the papers in our survey claim that the vast amount of imprecise data generated by NL users may provide tremendous benefits to software development organizations if processed with an NLP system. Most of the articles also indicate that NLP use with RE is still in its early stage; however, this research topic is rapidly expanding. Although NLP can compensate for many of the requirements’ ambiguity, inconsistency, and incompleteness, there are still circumstances where interaction with end-users is required for clarification [58]. This finding is supported by entries in the “User Interaction” column in Table I. Most of the papers about the use of NLP in requirements elicitation and analysis indicate that parsing requirement texts and classifying the information stored in them are difficult for humans. Thus, these activities should be automated as much as possible.

Furthermore, we discovered that most of the analyzed studies obtained their requirement datasets from external sources (e.g., Twitter, or an app store) rather than from existing documents for requirement elicitation. We found that the current state of the art in this area indicates that the first two phases prioritize researchers and practitioners. Table I shows most of the NLP techniques employed parsers and taggers to explain the tools used in text processing. This finding is supported by the results from [53]. In that study, NLP techniques are applied to sentence segmentation, tokenization, PoS tagging, shallow parsing, dependency parsing, word stemming, lemmatization, and role labeling.

Within the RE field, NLP is primarily used to analyze requirements and schedule them for further processing. This is primarily focused on developing models from elicited requirements and improving the consistency of the SRS. Both SRS and NLP share three common tasks: PoS tagging, rule-based analysis, and syntactic parsing. Two areas that lag behind the other RE (sub-)phases in the scope are requirements documentation (e.g., drafting of the SRS) and requirement prioritization. We assume that the writing requirements are supported by the NLP, including implementation of a specific template, spell checking, and explicitly resolving any possible ambiguities. These writing requirements may be beneficial to requirements engineers. However, since these tools necessitate live contact with requirement engineers, this research is not regularly performed than less advanced tools (e.g., solely for ambiguity checking).

Classification, model extraction, and detection seem to have more advancement than other areas of research, based on the number of publications and reported NLP tasks. This assumption is also valid for ambiguity detection.

VIII. Conclusion

This paper reviewed the current status of using NLP and its limitations in requirements elicitation and analysis processes. With the need for faster speed, lower cost, and higher quality in software engineering, there is an increasing need for automated support for all processing requirements of elicitation and analysis in the RE artifacts. While the pressure from industrial customers is obvious, extensive work is still needed to create automated NLP-based processes in RE. NLP tools and techniques have been proposed to automatically or semi-automatically detect syntactic, semantic, and pragmatic ambiguities in the requirements. Many solutions have been proposed from academia and industry to evolve the use of NLP in RE. Despite progress in NLP, there are still limitations to the NLP system. There is extreme pressure from the industry to improve the accuracy of NLP used to extract system requirements. Therefore, NLP must be enhanced for the elimination of residual errors. The accuracy must be substantially increased if it is to be seriously considered for use in RE. Based on the results from this paper's review and due to its limitations, NLP systems cannot be considered as a solution that can fix all RE issues. Nevertheless, NLP can be used to assist RE analysts. Findings from this study indicate that NLP can be used in real-world applications. Future research may produce more specialized NLP tools that can help consolidate the domain model and serve as translators between different RE documents and structured models.

These findings provide an understanding of the state of the art in this field of study and are useful for developing an analytical framework for complete systematic literature reviews. From the standpoint of RE, it may be important to investigate how NLP task combinations can be streamlined to fully perform additional tasks. The same is true for NLP in requirements management where requirements are managed within a software system, from elicitation to implementation to reuse. The NLP tasks described in this paper is not a comprehensive list. A possible extension of this research may examine NLP activities that have not yet been used in the field and how they may be used in the future applications.

Acknowledgment

The authors wish to thank the Public Authority for Applied Education and Training for providing the resources needed to conduct this research.

References

[1] T. Ambreen, N. M. Ikram, M. Usman, and M. Niazi, “Empirical research in requirements engineering: trends and opportunities,” Requirements Engineering, vol. 23, pp. 63–95, 2018.

[2] R. Vlas, and W. N. Robinson, “A rule-based natural language technique for requirements discovery and classification in open-source software development projects” 44th Hawaii Intl Conf. on Syst. Sci., vol. 2011, pp. 1–10, 2011.

[3] C. Huyck, and F. Abbas, “Natural language processing and requirements engineering: a linguistics perspective” in Proc 1st Asia Pac. Conf. on Software Quality, 2000.

[4] E. D.Liddy, Natural Language Processing. Encyclopedia of Library and Information Science, 2001.

[5] B. Kitchenham, Procedures for Performing Systematic Reviews. Keele, UK: Keele University, 2004, pp.1-26.

[6] Ö Albayrak, and J. C. Carver, “Investigation of individual factors impacting the effectiveness of requirements inspections: a replicated experiment,” Emp Softw Eng, vol. 19, pp. 241-266, 2014.

[7] E. Dubois, J. Hagelstein, and A. Rifaut, Formal Requirements Engineering. “with ERAE,” Philips J Res, vol. 43, pp. 393-414, 1988.

[8] D. Zowghi, and C. Coulin, “Requirements elicitation: A survey of techniques, approaches, and tools” in Eng Manag Soft Requirements. Springer, pp. 19-46, 2005.

[9] F. D. Milsom, “Using CORE with SDL to specify requirements” in Proc. Seventh Intl. Conf. on Software Engineering for Telecommunication Switching Systems: SETSS 89 (Conf. Publ. No. 306), vols. 137-141. London: Institution of Electrical Engineers (IEE), 1989.

[10] N. A. Ernst, and J. Mylopoulos, “On the perception of software quality requirements during the project lifecycle,” Lecture Notes in Computer Science. Berlin Heidelberg: Springer, pp. 143–157, 2010.

[11] Z. S. H. Abad, A. Shymka, S. Pant, A. Currie, and G. Ruhe, “What are practitioners asking about requirements engineering? an exploratory analysis of social q a sites” in 24th IEEE Int. Requirements Engineering Conf. Workshops (REW), pp. 334–343, 2016.

[12] W. Hu, J. C. Carver, V. K. Anu, G. S. Walia, and G. Bradshaw, “Detection of requirement errors and faults via a human error taxonomy: a feasibility study” in Proc. 10th ACM/IEEE Intl. Symp. on Empirical Software Engineering and Measurement, pp. 1-10, 2016, Sept.

[13] D. W. Cordes, and D. L. Carver, “Evaluation method for user requirements documents,” Information and Software Technology, vol. 31, pp. 181-188, 1989.

[14] I. Sommerville, and P. Sawyer, Requirements Engineering: A Good Practice Guide, Wiley, 1997.

[15] U. S. Shah, and D. C. Jinwala, "Resolving ambiguities in natural language software requirements: A comprehensive survey. ACM SIGSOFT Software Engineering Notes," vol. 40, pp. 1–7, 2015.

[16] G. C. Belev, "Guidelines for specification development" in Proceedings., Annual Reliability and Maintainability Symposium (pp. 15-21). IEEE, 1989, Jan.

[17] M. G. Christel, and K. C. Kang, 1992, Issues in Requirements Elicitation (No. CMU/SEI-92-TR-12). Pitts Burgh, PA: Software Engineering Institute, Camegie Mellon University.

[18] D. Firesmith, "Common requirements problems, their negative consequences, and the industry best practices to help solve them," J Object Technol, vol. 6, pp. 17-33, 2007.

[19] D. M. Berry, E. Kamsties, and M. M. Krieger, 2003, From Contract Drafting to Software Specification: Linguistic Sources of Ambiguity. Univ. of Waterloo, Tech. Rep.

[20] A. Lash, K. Murray, and G. Mocko, ""Natural language processing applications in requirements engineering." ASME 2012 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference." American Society of Mechanical Engineers, 2012.

[21] P. Rayson, R. Garside, and P. Sawyer, Language Engineering for the Recovery of Requirements from Legacy Documents, REVERE project report. Lancaster University, May 1999, 1999.

[22] S. Delisle, K. Barker, and I. Biskri, 1999, Object-Oriented Analysis: Getting Help from Robust Computational Linguistic Tools. na.

[23] P. Sawyer, P. Rayson, and R. Garside, "REVERE: support for requirements synthesis from documents," Information Systems Frontiers, vol. 4, pp. 343-353, 2002.

[24] G. G. Chowdhury, "Natural language processing," Annu Rev Inf Sci Technol, vol. 37, pp. 51-89, 2003.

[25] Tj. Eggebraaten, "Rj Stevens, and Ew will. Natural language processing ('NLPoverview')". US Patent 8,639,495, pp. 1–19, 2014.

[26] B. Langner, S. Vogel, and A. Black, Evaluating a Dialog Language Generation System: Comparing the MOUNTAIN System to Other NLG Approaches, 2010, pp. 1109–1112.

[27] R. Perera, and P. Nand, "Recent advances in natural language generation: A survey and classification of the empirical literature," Comput Inform, vol. 36, pp. 1–32, 2017.

[28] L. Zhao, W. Alhoshan, A. Ferrari, K. J. Letsholo, M. A. Ajagbe, E. Chioasca, and R. T. Batista-Navarro, Natural Language Processing (NLP) for Requirements Engineering: A Systematic Mapping Study. Available at: arXiv:2004.01099, 2020.

[29] B. Arendse, A thorough comparison of NLP tools for requirements quality improvement [Master's thesis], 2016.

[30] Euiwhan Kim Gi-Name Wang (eds.) Bhabani Shankar Prasad Mishra, Satchidananda Dehuri. Techniques and Environments for Big Data Analysis: Parallel, Cloud, and Grid Computing. Studies in Big Data 17. Springer International Publishing, 1st edition, 2016.

[31] A. Reshamwala, D. Mishra, and P. Pawar, "Review on natural language processing," IRACST Engineering Science and Technology: an International Journal (ESTIJ), vol. 3, pp. 113–116, 2013.

[32] D. Yuret, and F. Türe, "Handbook of natural language processing,", Proc. Main Conf. on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics -, pp. 328–334, 2006.

[33] F. Mokammel, E. Coatanéa, J. Coatanéa, V. Nenchev, E. Blanco, and M. Pietola, "Automatic requirements extraction, analysis, and graph representation using an approach derived from computational linguistics," Sys Eng, vol. 21, pp. 555-575, 2018.

[34] K. Ryan, "The role of natural language in requirements engineering". IEEE Intl Symp. on Requirements Eng., pp. 240–242, 1993.

[35] F. Fabbrini, M. Fusani, V. Gervasi, S. Gnesi, and S. Ruggieri, "Achieving quality in natural language requirements,", Proc. Int. SW Quality Week, pp. 4–5, 1998.

[36] O. Laitenberger, C. Atkinson, M. Schlich, and K. El Emam, "An experimental comparison of reading techniques for defect detection in UML design documents," J Sys Soft, vol. 53, pp. 183–204, 2000.

[37] L. Mich, and R. Garigliano, "Ambiguity measures in requirements engineering,", Proc. ICS 16th IFIP WCC, pp. 39–48, 2000.

[38] U. Ahmed, "A review of knowledge management in requirements engineering". Intl Conf. on Eng. and Emerg. Technol. (ICEET), pp. 1–5, 2018.

[39] S. Besrour, L. B. A. Rahim, and P. D. D. Dominic, "A quantitative study to identify critical requirement engineering challenges in the context of small and medium software enterprise" in 3rd Intl Conf. on Comput. and Inf. Sci. (ICCOINS), vol. 2016. IEEE, pp. 606-610, 2016, Aug.

[40] M. Bano, "Addressing the challenges of requirements ambiguity: a review of empirical literature" in 5th Intl Workshop on Empirical Requirements Eng. (EmpiRE), vol. 2015. IEEE, pp. 21-24, 2015.

[41] M. S. Murtazina, and T. V. Avdeenko, "An ontology-based approach to support for requirements traceability in agile development," Procedia Computer Science, vol. 150, pp. 628-635, 2019.

[42] Y. Lee, and W. Zhao, "An ontology-based approach for domain requirements elicitation and analysis,", Proc. First International Multi-Symposiums on Computer and Computational Sciences, pp. 364–371, 2006.

[43] G. I. Omoronyia, T. G. Sindre, S. T. Stålhane, T. S. Biffl, T. Moser, and W. Sunindyo, "A domain ontology building process for guiding requirements elicitation, International working conference on requirements engineering: foundation for software quality." Springer, pp. 188–202, 2010.

[44] M. G. Ilieva, and O. Ormandjieva, "Automatic transition of natural language software requirements specification into formal presentation" in, Lecture Notes in Computer Science, pp. 392–397, 2005.

[45] L. Mich, "NL-OOPS: from natural language to object oriented requirements using the natural language processing system LOLITA," Natural Language Engineering, vol. 2, pp. 161–187, 1996.

[46] K. Li, R. G. Dewar, and R. J. Pooley, 2004, Requirements Capture in Natural Language Problem Statements. Heriot-Watt Technical Report HW-MACS-TR-0023.

[47] D. K. Deeptimahanti, and M. A. Babar, "An automated tool for generating UML models from natural language requirements" in Proc. 2009 IEEE/ACM intl conf. on autom. softw. eng., pp. 680–682, 2009.

[48] M. Ibrahim, and R. Ahmad, "Class diagram extraction from textual requirements using natural language processing (NLP) techniques". Second International IEEE Conf. on Comput. Res. and Dev., pp. 200–204, 2010.

[49] W. Maalej, M. Nayebi, T. Johann, and G. Ruhe, "Toward data-driven requirements engineering," IEEE Software, vol. 33, pp. 48–54, 2015.

[50] E. C. Groen, J. Doerr, and S. Adam, "Towards crowd-based requirements engineering a research preview," Lecture Notes in Computer Science, pp. 247–253, 2015.

[51] M. Robeer, G. Lucassen, J. M. E. M. van der Werf, F. Dalpiaz, and S. Brinkkemper, "Automated extraction of conceptual models from user stories via NLP. IEEE" 24th International Requirements Engineering Conf. (RE), pp. 196–205, 2016.

[52] L. H. Bordignon, T. S. Thom, V. Silva, S. Dani, M. Fantinato, and R. C. B. Ferreira, "Natural language processing in business process identification and modeling: A Systematic Literature Review," Proc. XIV Braz. Symp. on Information Systems, pp. 1-8, 2018.

[53] G. Lucassen, M. Robeer, F. Dalpiaz, J. M. E. M. van der Werf, and S. Brinkkemper, "Extracting conceptual models from user stories with Visual Narrator," Requirements Engineering, vol. 22, pp. 339-358, 2017.

[54] C. Hollis, and T. Bhowmik, "Automated support to capture verbal justin-time requirements in agile development: A practitioner view" in 25th International Requirements Engineering Conf. Workshops (REW). IEEE, pp. 419–422, Sept 2017.

[55] E. C. Groen, S. Kopczyńska, M. P. Hauer, T. D. Krafft, and J. Doerr, "Users—the hidden software product quality experts?: A study on how app users report quality aspects in online reviews" in 25th International

Requirements Engineering Conf. (RE), vol. 2017. IEEE, pp. 80-89. IEEE, 2017, Sept.

[56] B. D. Cruz, “Detecting vague words & phrases in requirements documents in a multilingual environment. IEEE” 25th International Requirements Engineering Conf., pp. 233–242, 2017.

[57] C. S. R. K. Surana, D. B. Gupta, and S. P. Shankar, “Intelligent chatbot for requirements elicitation and classification” in 4th Intl Conf. on Recent Trends on Electron., Information, Communication & Technol. (RTEICT), vol. 2019. IEEE, pp. 866-870, 2019, May.

[58] F. S. E. Friesen, F. S. Bäumer, M., and M. Geierhos, “CORDULA: software requirements extraction utilizing chatbot as communication interface,” REFSQ Workshops, vol. 2075, 2018.

[59] J. Slankas, and L. Williams, “Automated extraction of non-functional requirements in available documentation.” 1st Intl Workshop on Nat. Lang. Anal. in Softw. Eng. (NaturaLiSE), pp. 9-16, May 2013.

[60] R. Veleda, and L. M. Cysneiros, “Towards an ontology-based approach for eliciting possible solutions to non-functional requirements” in, Lecture Notes in Computer Science Intl Conf. on Adv. Inf. Syst. Eng. Cham: Springer, (pp. 145-161), 2019, Jun.

[61] H. Yang, and P. Liang, “Identification and classification of requirements from app user reviews. EASE,”, Proc. 21st Intl. Conf. on Evaluation and Assessment in Software Engineering, June (2017), pp. 344–353, '17.

[62] M. Lu, and P. Liang, “Automatic classification of non-functional requirements from augmented app user reviews, ”, Proc. 21st Intl. Conf. on Evaluation and Assessment in Software Engineering, pp. 344-353, 2017.

[63] M. S. Murtazina, and T. V. Avdeenko, “Ontology-based approach to the requirements engineering in agile environment” in International Scientific-Technical Conference on Actual Problems of Electronics Instrument Engineering (APEIE), vol. 2018 XIV. IEEE, pp. 496-501, 2018, Oct.

[64] S. Gnesi, and G. Trentanni, “QuARS: a NLP tool for requirements analysis” in REFSQ Workshops, 2019.

[65] M. Mezghani, Juyeon Kang, and F. Sèdes, “Using k-Means for Redundancy and Inconsistency Detection: application to Industrial Requirements ” in Natural Language Processing and Information Systems, M. Silberztein et al, Eds. Cham: Springer International Publishing. ISBN: 978-3-319-91947-8, pp. 501–508, 2018.

[66] F. Dalpiaz, A. Ferrari, X. Franch, and C. Palomares, “Natural language processing for requirements engineering: the best is yet to come,” IEEE Software, vol. 35, pp. 115-119, 2018a.

[67] K. Lee, L. He, M. Lewis, and L. Zettlemoyer, “End-to-end neural coreference resolution” in Proc. 2017 Conf. on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, pp. 188–197, 2017.

[68] H. Van der Aa, J. Carmona Vargas, H. Leopold, J. Mendling, and L. Padró, “Challenges and opportunities of applying natural language processing in business process management. In COLING 2018: the,”, Proc. Conf.: August 20-26, 2018 Santa Fe, New Mexico, USA 27th Intl Conf. on Comp. Linguist. Association for Computational Linguistics, pp. 2791-2801, 2018.

[69] F. A. Acheampong, C. Wenyu, and H. Nunoo - Mensah, “Text-based emotion detection: advances, challenges, and opportunities,” Engineering Reports, vol. 2, p. e12189, 2020.

[70] C. Guo, J. Cao, X. Zhang, K. Shu, and H. Liu, 2019, Dean: Learning Dual Emotion for Fake News Detection on Social Media. arXiv Preprint ArXiv:1903.01728.

[71] P. Zhong, D. Wang, and C. Miao, “Knowledge-enriched transformer for emotion detection in textual conversations, CoRR abs/1909.10681 (2019). URL”: http://arxiv, org/abs/1909.10681.

[72] A. Seyeditabari, N. Tabari, S. Gholizadeh, and W. Zadrozny, 2019, Emotion Detection in Text: Focusing on Latent Representation. arXiv Preprint ArXiv:1907.09369.

[73] M. Abdul-Mageed, and L. Ungar, “Emonet: fine-grained emotion detection with gated recurrent neural networks” in Proc. 55th Annu. Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 718-728, 2017, Jul.

[74] H. Rashkin, E. M. Smith, M. Li, and Y. L. Boureau, “Towards empathetic opendomain conversation models: a new benchmark and dataset” in Proc. 57th Annu. Meeting of the Association for Computational Linguistics, Florence, Italy. Association for Computational Linguistics, pp. 5370–5381, Jul. 2019.

[75] Z. Kaddari, Y. Mellah, J. Berrich, M. G. Belkasmi, and T. Bouchentouf, “Natural language processing: challenges and future directions” in Intl Conf. on Artif. Intell. & Ind. Appl. Cham: Springer, pp. 236-246, 2020, Mar.

[76] B. E. Stein, T. R. Stanford, and B. A. Rowland, “The neural basis of multisensory integration in the midbrain: its organization and maturation,” Hearing Research, vol. 258, pp. 4–15, 2009.

[77] G. Schirner, D. Erdogmus, K. Chowdhury, and T. Padir, “The future of human-in- the-loop cyber-physical systems,” Computer, vol. 46, pp. 36–45, 2013.

[78] Y. Srivastava, V. Murali, S. R. Dubey, and S. Mukherjee, 2019, Visual Question Answering Using Deep Learning: A Survey and Performance Analysis. arXiv Preprint ArXiv:1909.01860.

[79] P. Kordjamshidi, T. Rahgooy, and U. Manzoor, “Spatial language understanding with multimodal graphs using declarative learning based programming” in Proc. 2nd Workshop on Structured Prediction for Natural Language Processing, Copenhagen, Denmark. Association for Computational Linguistics, pp. 33–43, Sept. 2017.

[80] R. Sanabria, O. Caglayan, S. Palaskar, D. Elliott, L. Barrault, L. Specia, and F. Metze, 2018, How2: a Large-Scale Dataset for Multimodal Language Understanding. arXiv Preprint ArXiv:1811.00347.

[81] M. S. Amac, S. Yagcioglu, A. Erdem, and E. Erdem, “Procedural reasoning networks for understanding multimodal procedures,” Arxiv e-Prints, arXiv-1909, 2019.

[82] Z. Yu, Y. Cui, J. Yu, D. Tao, and Q. Tian, 2019, Multimodal Unified Attention Networks for Vision-and-Language Interactions. arXiv Preprint ArXiv:1908.04107.

[83] H. van der Aa, H. Leopold, and H. A. Reijers, “Dealing with behavioral ambiguity in textual process descriptions,”, Lecture Notes in Computer Science. Intl Conf. on Bus. Process Manag. Springer International Publishing, pp. 271–288, 2016.

[84] F. Dalpiaz, I. Van der Schalk, and G. Lucassen, “Pinpointing ambiguity and incompleteness in requirements engineering via information visualization and NLP” in, Lect Notes Comput Sci. Cham: Springer, (pp. 119-135), 2018b, Mar.

[85] T. Iqbal, P. Elahidoost, and L. Lucio, “A bird's eye view on requirements engineering and machine learning” in 25th Asia-Pacific Software Engineering Conf. (APSEC), vol. 2018. IEEE, pp. 11-20, 2018, Dec.

[86] N. Madhavji, L. Pasquale, A. Ferrari, and S. Gnesi, Eds., “Requirements engineering: foundation for software quality”: 26th International Working Conf., REFSQ 2020, Pisa, Italy, March 24–27, 2020, 2020, Proceedings (Vol. 12045). Springer Nature.

[87] H. Femmer, “Requirements quality defect detection with the Qualicen requirements scout” in REFSQ Workshops, 2018.

[88] A. Al-Hunaiyyan, R. Alhajri, A. Alzayed, B. Alraqqas. “Towards an effective Distance Learning Model: Implementation Framework for Arab Universities”. International Journal of Computer Applications. Vol. 6, Issue 5, 2016.

[89] A, Bimba, N. Idris, R. Mahmud, A. Al-Hunaiyyan. “A Cognitive Knowledge-based Framework for Adaptive Feedback”. In S. Phon-Amnuaisuk, T. W. Au & S. Omar (Eds.), Computational Intelligence in Information Systems; Proceedings of the Computational Intelligence in Information Systems Conference (CIIS 2016). Pp. 245-255. Cham: springer International Publishing.

[90] R. Alhajri, S. Al-Sharhan. A. Al-Hunaiyyan, T. Al-Othman. “Design of educational multimedia interfaces: individual differences of learners”. Proceedings of (KCESS ’11) the Second Kuwait e-Services and e-Systems Conference. April 5-7, 2011. Kuwait.