Journal of Universal Computer Science, vol. 28, no. 11 (2022), 1136-1168
submitted: 2/12/2021, accepted: 24/9/2022, appeared: 28/11/2022 CC BY-ND 4.0

Building an integrated requirements engineering process based on Intelligent Systems and Semantic Reasoning on the basis of a systematic analysis of existing proposals

Alexandra Corral
(University of the Armed Forces ESPE, Sangolquí, Ecuador
https://orcid.org/0000-0002-9225-1346, macorral@espe.edu.ec)

Luis E.Sanchez
(GSyA Research Group, University of Castilla-La Mancha, Ciudad Real, Spain
https://orcid.org/0000-0003-0086-1065, luise.sanchez@uclm.es)

Leandro Antonelli
(Lifia – Faculty of Computer Science, National University of La Plata, La Plata, Argentina
https://orcid.org/0000-0003-1388-0337, lanto@lifia.info.unlp.edu.ar)

Abstract: Requirements Engineering is one of the fundamental activities in the software development process and is oriented toward what should be produced. One of the development team's most common problems is a lack of communication regarding an understanding of the discourse domain and how to integrate and process excessive information originating from different sources. This may lead to errors of omission and the consequent production of incomplete and inconsistent artifacts, which will have a direct effect on the quality of the software. The use of machine learning techniques helps the development team produce successful software on the basis of the acquisition of knowledge and human experience with which to understand the domain of the application. This paper, therefore, presents a proposal for a new methodological process oriented toward the construction of a vocabulary concerning the application domain. The authors propose to do this by employing Natural Language Processing (NLP), ontologies and heuristics that will lead to the production of a Lexicon that is common to analysts and customers, both of whom will understand the universe of discourse, thus mitigating problems of completeness. This objective has been achieved by carrying out a Systematic Literature Review of the artificial intelligence techniques employed in the requirements engineering process, which led to the discovery that 41.37% use NLP, while 55.71% apply ontologies such as semantic reasoners which help solve the problem of language ambiguity, the structures in specifications or the identification of key concepts with which to establish traceability links. However, the review also showed that the problems regarding the comprehension and completeness of requirements problems have yet to be resolved.

Keywords: Software Engineering; Software Requirements; Intelligent Systems Artificial Intelligence Techniques; Natural Language Processing; Semantic Reasoning.
Categories: D.2.0, D.2.1, M.8
DOI: 10.3897/jucs.78776

1 Introduction

Software development is a multifaceted task in which new challenges frequently appear [Basirati et al., 2020], in addition to being an intensive activity as regards knowledge and collaboration [Nunes et al., 2020)]. The highly complex part of software development depends, to a great extent, on the initial phases, such as: the analysis and the recompilation of requirements in order to produce correct design artifacts [Yang et al., 2018].

Requirements engineering involves the association of four stages: elicitation, analysis, specification and validation [Andrews et al., 2014], including the management of requirements [Laplante, 2017], which is a continuous process that lasts for the duration of the whole project. These activities are considered to be traditional methodologies.

However, the appearance of agile methodologies led to the term “agile engineering requirements”, and change management is one of the principles in software development, signifying that requirements are dealt with during the useful life of a piece of software [Curcio et al., 2018; Urbieta et al., 2020].

The objective of the requirements engineering process is to translate the descriptions of the requirements into semi-structured documents [Landhäußer et al., 2014], which are then translated into implemented specifications for software providers [Lee et al., 2016], thus facilitating the generation of conceptual models (Bozyiğit et al., 2021)], design and program architectures. The requirements are a verbalization of the alternatives of a decision regarding the functioning and quality of a system [Maalej et al., 2015], and are obtained on the basis of natural language [Zhao et al., 2021] that is distorted into textual expressions [Dick et al., 2017], and this factor can make it difficult to represent the requirements of the system. According to Jackson in [Jureta et al., 2008], the problem of the requirements concerns the search for the specifications (S), which for suppositions of the domain (K) satisfy the requirements (R). These requirements are generally described as incomplete, ambiguous and inconsistent, and this directly affects the whole software development process [Zait and Zarour, 2018].

Human-rooted conflicts are detrimental to the success of software projects [Basirati et al., 2020]. From the point of view of the development equipment, factors such as: the lack of coordination, communication barriers, little comprehension of the domain, excessive management of information, etc., directly affect the requirements analysis and the decision-making process [Annaiahshetty and Prasad, 2013]. The requirements engineering process requires a great human effort, although case tools are very useful as regards providing support to the process. Human and intellectual activities are required in order to analyze and verify the valid elements that are produced during the first stages [Vasanthapriyan et al., 2015].

The application of non-conventional technologies originating from artificial intelligence, which imitate behavior similar to that of human intelligence, could benefit the development team as regards solving common problems in requirements engineering [Ninaus et al., 2014], as could intelligent techniques that explore data for knowledge discovery, reasoning, learning, planning, natural language processing, perception or supporting decision-making [Perkusich et al., 2020].

Some research has proposed solutions in which artificial intelligence is superimposed onto requirements engineering through the use of intelligent systems

[Fernández-Montes et al., 2014] that are categorized by cognitive reasoning such as: expert systems [Gupta and Nagpal, 2020], recommendation systems [AlZu'bi et al., 2018], intelligent agents [Clark et al., 2021], case-based reasoning (CBR) [Kaur et al., 2021]etc., with the purpose of giving intelligent support to the requirements engineering process, but little research deals with the possibility of supporting the aspects of analysis and human reasoning [Jin Guo et al., 2014].

The representation of knowledge in an artificial intelligence program implies selecting a series of conventions with which to describe all the aspects that can be computationally represented, and these are described as ontologies [Gayathri and Uma, 2018]. They are used as a basis for the knowledge within the intelligent systems, and incorporate semantic rules that can infer human reasoning, thus assisting in decision making during the initial phases of software development [Preece, 2018].

Those requirements engineering process approaches that have been fused with artificial intelligence have shown promising results, although they have still not been exploited [Kaur et al., 2021; Kulkarni et al., 2021], signifying that more empirical studies are necessary. There is, however, a growing interest in these related subjects, as evidenced by their increasing appearance in conferences, such as: the International Requirements Engineering Conference (RE), the International Scientific and Technical Conference on Computer Sciences, the International Conference on Progress in Informatics and Computing (PIC), the Proceedings from the International Conference on Information Integration and Web-based Applications & Services (IIWAS '13), the International Conference on Software Engineering (ICSE), the International Conference on Advances in Computing, Communications and Informatics (ICACCI), and the International Conference on Information Technology and Nanotechnology (ITNT). Several current studies have related subjects of interest in requirements engineering processes to artificial intelligence as regards the ambiguity of the definitions of requirements, reuse, traceability, planning and analysis. However, the systematic comparison of methods that will resolve problems related to communication, along with the coordination and understanding of the domain, and that are incorporated into the initial phase are challenges that have yet to be resolved. The objective of this paper is to fill this gap.

This paper specifically provides information regarding a systematic review of 29 current studies based on artificial intelligence techniques whose output is intelligent systems in requirements engineering processes. The objective of this review is to discover the various artificial intelligence techniques that have been used to classify the steps in these processes, how they work and how they are evaluated. These findings will make it possible to identify the ability of certain techniques to resolve existing challenges and to find weak phases in requirements engineering processes that have not yet been explored using artificial intelligence.

The process of carrying out systematic literature reviews consists of defining a strict sequence of methodological steps, which are developed according to a protocol. This is carried out on the basis of a central topic, which represents the nucleus of the investigation, and which can be expressed by the use of a specific, predefined, focused and structured question. Systematic literature reviews are usually carried out by following the guidelines of [Kitchenham and Brereton, 2013], which are appropriate for software engineering researchers.

This paper is organized as follows. Section 2 shows the research questions defined, while Section 3 provides an explanation of the review method, which is based on the review protocol in which the search strategy, selection criteria and execution of the papers are defined. Section 4 shows the results obtained, along with a discussion, while Section 5 presents the methodological process used to construct the vocabulary regarding the application, Section 6 show evaluation the proposed approach and show threats to validity. Finally, Section 7 shows the authors' conclusions and future work.

2 Research Questions

This section provides a definition of the objectives of this research [Kitchenham and Brereton, 2013]. The focus of the question identifies experiences, initiatives and reports on the input of intelligent systems to the requirements engineering process through the use of ontologies, and the concepts used for their construction.

The frequent problems with software development equipment, and specifically those that occur among requirements analysts, take place because of the lack of coordination, barriers to communication, not having sufficient comprehension of the domain, excessive management of information, etc., which directly affect analysis and decision making during the requirements engineering process.

Intelligent systems can help consulters develop successful software on the basis of the acquisition of knowledge and human experience, and that knowledge can be modeled through the use of ontologies.

The research question employed to carry out the research is the following:

Which semantic reasoning-based intelligent systems are used to improve requirements engineering software requirements?

The key words and synonyms of which this question is composed, and those that will be used during the execution of the review, are:

- Requirements Engineering: Software Engineering: Software Requirements, Requirements process, Requirements Elicitation.
- Intelligent Systems: Recommender Systems, Expert System, intelligent agent, Knowledge Base, Smart agent, Case-based reasoning
- Semantic: Ontology, ontologies, ontologies engineering, Semantic interoperability.
- Artificial Intelligence Techniques: Machine learning, Heuristics, Support Vector Machines, Artificial Neural Networks, Natural Language Processing

This research considers publications that have taken into account the application of artificial intelligence techniques in order to support requirements engineering activities through the use of semantic reasoners. The result expected at the end of this systematic review will be the identification of the initiatives related to the inputs provided by intelligent systems during the requirements engineering process through the use of ontologies, while the measurement of the result will be the number of initiatives identified. The principal areas of application that will benefit from the results of this systemic review are software development, requirements experts, knowledge engineers and software engineers.

3 Review method

An SLR is a review system that employs systematic approaches in order to identify, determine and evaluate relevant research. The review process employed herein is based on the protocol proposed by [Kitchenham and Brereton, 2013].

A systematic review is developed using the following structure: Selection of Sources, Search String, Selection of studies, Quality evaluation of the studies selected and Extraction of Information, after which the Results and a Discussion are presented.

The methods used in each stage of the systematic review are described in the remainder of this section.

3.1 Selection of Sources

The objective of this subsection is to show the sources used to search for the primary studies that would be analyzed in this review.

The search for primary studies is carried out by employing web search motors, electronic databases and manual searches, such as: research in magazines, conferences, specific books or in research publications suggested by experts on the subject.

The principal sources in which systemic reviews are carried out are: Scopus, Springer, ACM Digital Library, IEEE Xplore, Google Scholar, Springer, ScienceDirect.

3.2 Search String

A search string has the objective of capturing all results that relate to the subject being studied. In this case, we sought results related to requirements engineering linked to artificial intelligence using semantics.

The search string was defined according to the PICOC criteria (population, intervention, comparison, result and context) recommended by [Kitchenham et al., 2007), as shown in Table 1.

Population	Intervention	Comparison	Outcome	Context
Requirements Engineering	Artificial Intelligence Techniques in requirements engineering	Not applicable	A comparison of the artificial intelligence techniques in requirements engineering processes whose output is intelligent systems.	Requirements Engineering

Table 1: PICOC criteria used to define the search string

The terms were connected using Boolean operators; the operator OR was employed for synonyms (alternative terms), while AND was utilized in order to link the search. This led to the following search string:

> *(“Requirements engineering” OR “Software requirements” OR “Requirements process“) AND “Artificial Intelligence Techniques” AND (Semantic OR Ontology).*

3.3 Selection of Studies

Once the sources have been defined, it is necessary to describe the process, the selection criteria and the evaluation of the studies in order to reduce the probability of bias. The selection criteria are decided during the definition of the protocol. The inclusion and exclusion criteria must be based on the research question.

The studies consequently had to present new initiatives (published no longer than 9 years before the search took place) that considered artificial intelligent techniques in the requirements engineering process, incorporating semantic reasoning for the construction of intelligent systems. The research had to employ comparative elements, methods, tools and techniques that could support the research question.

In order to select the initial set of studies, the title and abstract of all the initiatives obtained were read and evaluated according to all the inclusion and exclusion criteria. This initial set of studies was then further narrowed down by reading the complete texts.

The study selection process was carried out by all three researchers by means of critical reading. This process led to the exclusion of duplicated papers and those that did not fulfill the inclusion criteria or did not fit the context of the research. Quality criteria were then established in order to include only those papers that were relevant to the research.

The systematic review process and the number of papers identified in each stage are shown in Figure 1.

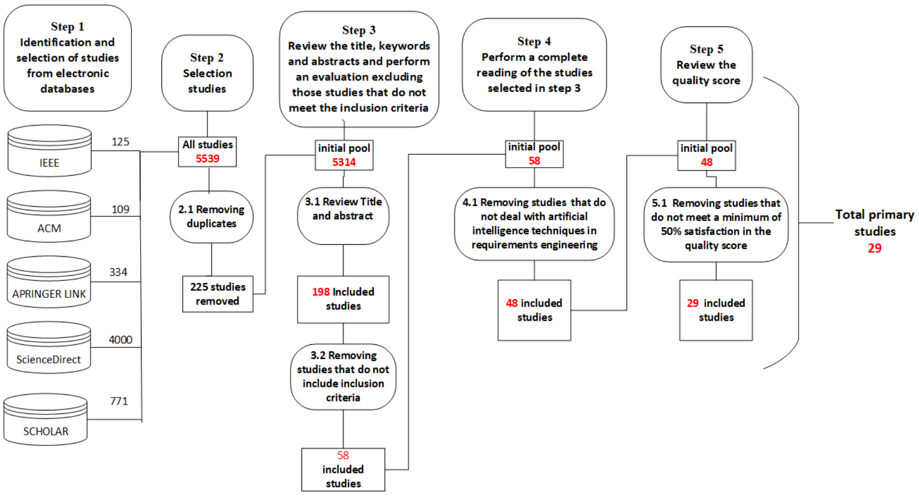
Step 1
Identification and selection of studies from electronic databases
Step 2
Selection studies
Step 3
Review the title, keywords and abstracts and perform an evaluation excluding those studies that do not meet the inclusion criteria
Step 4
Perform a complete reading of the studies selected in step 3
Step 5
Review the quality score
IEEE
125
ACM
109
APRINGER LINK
334
ScienceDirect
4000
SCHOLAR
771
All studies
5539
2.1 Removing duplicates
225 studies removed
initial pool
5314
3.1 Review Title and abstract
198 Included studies
3.2 Removing studies that do not include inclusion criteria
58
included studies
initial pool
58
4.1 Removing studies that do not deal with artificial intelligence techniques in requirements engineering
48 included studies
initial pool
48
5.1 Removing studies that do not meet a minimum of 50% satisfaction in the quality score
29 included studies
Total primary studies
29

Figure 1: Systematic review process

3.4 Quality evaluation of the selected studies

Each of the 48 studies filtered out in Stage 4 (subsection 3.3) were evaluated by each of the members of the research team using the Kappa index according to 8 quality criteria. The quality (QA) of the works selected was evaluated using a branding technique [Dermeval et al., 2016]. The evaluation instrument is presented in Table 1.

The criteria were classified in two categories. The first category was evaluated using the three objective questions (Q1,Q2, Q5 and Q8) in accordance with the scope of the research question, while the second category was evaluated using four subjective questions (Q3, Q4, Q6 and Q7) adapted from other literature reviews [Dybå and Dingsøyr, 2008; Vasanthapriyan et al., 2015] and that covered aspects of quality that should be taken into consideration when evaluating the studies identified in the review, such as:

- *Rigor.* Is the research approach complete and appropriate as regards attaining the objectives of the research?
- *Credibility.* Are the findings well presented and significant?
- *Relevance.* Do the findings discovered provide value to the research or for industrial practice and the research community?

The scoring was determined using a two-point measurement scale (Yes/No) for criteria Q1, Q6 and Q7, where Y=1 and N=0, and where 1 is a high contribution and 0 is zero, while the criteria Q2, Q3 and Q4 were additionally measured using the value of Partially (P= 0.5) if the contribution was weak.

Question Q5 considers the activities in the requirements engineering process that include management as part of the process, and this is determined in the relation Phases Number/Total Phases [Dermeval et al., 2016]. Criterion Q5 was evaluated independently, since it depends on the context of the study.

In this evaluation, all the studies that obtained a response of at least "0" for criteria Q3, Q4 and Q7 were excluded because they required a minimum threshold for this review. The remaining criteria ensured that the research contained sufficient information to be able to carry out analyses and comparisons in the proposed areas.

Once the quality evaluation had been applied, 19 studies were excluded, as shown in stage 4.1 of Figure 1, signifying those 29 studies remained for analysis and the extraction of the results shown in Table 3.

	Question	Possible Answer
1	Does it use artificial intelligence techniques?	Y=1, N=0
2	Is there a clear description of the algorithms used for the intelligent systems?	Y=1, N=0, P=0.5
3	Is there a clear definition of the objectives?	Y=1, N=0, P=0.5
4	Is there an adequate description of the context? (Industry, laboratory, products employed)	Y=1, N=0, P=0.5
5	Which RE process supports the study?	NumOfPhases/ TotalOfPhases
6	Is the study supported by a tool?	Y=1, N=0
7	Does the research add value to the Industrial? Community?	Y=1, P=0.5
8	Do the studies apply ontologies as a knowledge base in the intelligent systems?	Y=1, N=0

Table 2: Quality evaluation criteria

3.5 Extraction of Information

Having defined the search and selection processes, the data was extracted by reading the introduction, conclusions and a selection of the complete text of each of the works selected.

The information extracted from the studies had to contain methods, techniques, processes or strategies regarding artificial intelligence techniques based on semantic reasoning that can solve requirements engineering problems.

The ontologies had to represent the knowledge of specific domains or general requirements engineering concepts, thus enabling a comparison with initiatives that can contribute to reasoning about the requirements engineering process.

This section provides a description of each of the selected studies, shown in the previous section, according to the information extracted. The scores attained for each of the research works are shown in Table 2:

- *S1 - Williams (2018) "An Ontology Based Collaborative Recommender System for Security Requirements Elicitation":* The authors propose a collaborative recommendation system based on an ontology with which to obtain security requirements and evaluate the performance of the system.
- *S2 – Hovorushchenko and Pavlova (2018) "Evaluating the Software Requirements Specifications Using Ontology-Based Intelligent Agent":* This work develops an intelligent agent based on ontologies (OBIA) that can evaluate the level of sufficiency of the information in the SRS and detect missing attributes through the creation of methods developed for the analysis.
- *S3 - Huaqiang et al. (2017) "The research of domain ontology recommendation method with its applications in requirement traceability":* This study proposes an ontology-based requirements tracking method. The ontologies are used as intermediate artifacts that can solve the coincidence problem of keywords in the semantic dimension for the tracking requirements.
- *S4 - Bargui et al. (2011) "A decision making ontology building process for analytical requirements elicitation":* The authors employ an object-oriented model as a basis on which to obtain analytic requirements. The formal aspect of the ontology allows the automatic reasoning of the knowledge in the domain during requirements elicitation.
- *S5 - Al-Hroob et al. (2018) "The use of artificial neural networks for extracting actions and actors from requirements document":* This paper describes the development of a semi-automatic approach, denominated as IT4RE, for the extraction of use cases (action, actors) in the description of system requirements based on natural language.
- *S6 - Di Noia et al. (2018) "A fuzzy ontology-based approach for tool-supported decision making in architectural design":* This proposal is based on the application of non-functional requirements (NFR) and design patterns for decision making to the architectonic solutions of the system, for which it presents a support system for decisions that is based on ontologies.
- *S7 - Wang (2015) "Semantic information extraction for software requirements using semantic role labeling":* The author focuses on the domain analysis for the software products line (SPL). The study proposes a new approach with which to automatically extract the semantic information from the software requirement specifications (SRS), using the combination of semantic role tagging techniques and domain knowledge models.

- *S8 - Z. Wang et al. (2019) "A novel data-driven graph-based requirement elicitation framework in the smart product-service system context":* The research proposed establishes a systematic framework with which to orient the requirements attainment process through the extraction of massive data generated by the user and detected by Smart Connected Products (SCPs) in the context of Smart Product-Service System (Smart PSS).
- *S9 - Coudert et al. (2012) "Case-based reasoning and system design: An integrated approach based on ontology and preference modeling":* This work proposes a case-based reasoning process (CBR) that is integrated in order to design the system. It defines an ontology, which helps the designers to formalize the knowledge in the case-based reasoning CBR process during the first phases of development.
- *S10 - Morandini et al. (2017) "Engineering requirements for adaptive systems":* The development provides modeling characteristics, which strengthen the requirements analysis as regards specific knowledge and the decision criteria that a system needs in order to adapt to the dynamic changes in an autonomous manner.
- *S11 - Boukhari et al. (2013) "Efficient, Unified, and Intelligent User Requirement Collection and Analysis in Global Enterprises":* A semantic and scalable focus is proposed, which unifies the vocabulary and formal traditional representations by means of ontologies in order to solve heterogeneity problems. It uses formalisms to represent the requirements and reason in an efficient manner.
- *S12 - J. Guo et al. (2017) "Semantically Enhanced Software Traceability Using Deep Learning Techniques":* This presents a solution that uses learning to incorporate the semantics of requirements artifacts and the domain knowledge into the tracking solution. It proposes a tracking network architecture that uses word-incrustation models and recurrent neural networks (RNN) to generate tracking links, thus allowing requirements artifacts to be traced in an automatic and semantic manner.
- *S13 - Zaitand Zarour, (2018) "Addressing Lexical and Semantic Ambiguity in Natural Language Requirements":* One of the problems that appears in requirements engineering is ambiguity, generally because the requirement has been expressed in natural language. The authors propose the detection and resolution of ambiguity problems before documenting, through the adoption of natural language processing (NLP), the semantic web technique and the measurement of the similarity between requirements.
- *S14 - Gill et al. (2014) "Semi-automation for ambiguity resolution in Open Source Software requirements":* The heterogeneity of the information originating from the development of open source code is another challenge, which the authors attempt to solve thorough the disambiguity of the requirements in the Open Source Software Development (OSSD) context; the authors consequently propose a framework with the combination of human knowledge and machine-produced algorithms (NLP, WSD, Text mining).
- *S15 - Felfernig et al. (2017) "OpenReq: recommender systems in requirements engineering":* The proposed approach is based on a technological recommendation and decision–making tool for requirements

engineering processes, based on 4 aspects: recommendations for groups, the detection of dependencies, and dealing with conflicts and management.

- *S16 - Singh et al. (2016) "Rule-based system for automated classification of non-functional requirements from requirement specifications":* The objective of this proposal is to minimize and facilitate processes in the identification of non-functional requirements, thus allowing the designer to make decisions about the design and quality of the SRS.
- *S17 - Hariri et al. (2014) "Recommendation Systems in Requirements Discovery":* The authors propose a recommendation system with which to discover requirements in discussion forums. The study is focused on two aspects: The first is based on the organization and administration of the users' interests as regards recommending threads of discussion about a specific topic, while the second is related to the compilation of existing data in public access networks for the construction of association rules.
- *S18 - Ninaus et al. (2014) "INTELLIREQ: Intelligent Techniques for Software Requirements Engineering":* These authors present two content-based recommendation approaches that can support the requirements engineering process. They first propose a keyword recommender in order to facilitate the reuse of requirements.
- *S19 - K. Annaiahshetty and Prasad (2013) "Expert System for Multiple Domain Experts Knowledge Acquisition in Software Design and Development":* The approach in this proposal is based on the prototype development of an expert system with which to provide support to the software developer in the development life cycle. The authors focus their study on the acquisition of knowledge from experts in multiple domains that can facilitate the definition and use of specific artifacts for software development, analyzing their advantages and disadvantages.
- *S20 - Hovorushchenko et al. (2019) "Ontology-Based Intelligent Agent for Determination of Sufficiency of Metric Information in the Software Requirements":* This research proposes the implementation of an ontology-based intelligent agent (OBIA) in order to determine metric information in software requirements. Some of the actions permitted by the intelligent agent are: Partially deleting the person from the information processing processes, avoiding the loss of information and enhancing the quantity of metric information in the requirements recompilation phase, or improving software quality.
- *S21 - Liu, (2016) "CDNFRE: Conflict detector in non-functional requirement evolution based on ontologies":* This author bases his research on the analysis of non-functional requirements (NFR) conflicts. A system is created that makes it possible to detect conflicts automatically during the evolution of the NFR, using the ontologies as semantic reasoners of relationships among requirements. Conflicts are detected by referring to a rules system, which allows the attainment of new knowledge concerning derivate requirements.
- *S22 - Emebo et al. (2016) "An automated tool support for managing implicit requirements using Analogy-based Reasoning":* The authors, therefore, propose an analysis for the identification and management of non-detected requirements in software projects. The approach provides a solution by means

of an infrastructure consisting of three technologies: Natural Language Processing (NLP) for the analysis of requirements declarations, Ontologies for the extraction of the domain knowledge, and ABR to discover unknown and non-obtained requirements, and the reuse of these requirements in a new domain by means of analogy.

- *S23 - Han (2015) "Discriminating risky software project using neural networks":* This analysis is focused on predicting the risk of software projects in early stages before implementation. The solution is provided by developing a Neural Network (NN) model, using an algorithm to reduce risks.
- *S24 - Jin Guo et al. (2014b). "Towards an intelligent domain-specific traceability solution":* The authors present solutions as regards the traceability of requirements, incorporating a knowledge base composed of action units, link heuristics and a domain ontology. They employ a Contextualized Intelligent Traceability in the Domain (DoCIT) system, which is able to perform human reasoning for the highly focused area of communication and control in a transport domain.
- *S25 - Ben Abdessalem Karaa et al. (2016) Automatic builder of class diagram (ABCD):* an application of UML generation from functional requirements": This approach concerns extracting a class model from the user's requirements, expressed in natural language. It deals with the semantic analysis of the pre-processed text for the construction of the relevant information in a conceptual model.
- *S26 – Murtazina and Avdeenko. (2020). "The ontology-driven approach to intelligent support of requirements engineering in agile software development":* The authors present an approach for the intelligent support of requirements during the development of agile software using a system of ontological models that integrate requirements engineering process information for the agile management of projects and the domain knowledge of the application to be developed.
- *S27 - (Do et al., 2020). "Capturing creative requirements via requirements reuse: A machine learning-based approach":* This paper presents an automated framework for the generation of creative requirements by reusing software requirements obtained online through the use of automatic learning techniques and natural language processing. The framework is constructed by recompiling product requirements that have been obtained online, carrying out a filtering process and grouping the resulting requirements in clusters.
- *S28 - (Adithya & Deepak, 2021). "OntoReq: An Ontology Focused Collective Knowledge Approach for Requirement Traceability Modelling":* These authors propose an architecture with which to generate a requirements traceability matrix based on knowledge engineering by using an ontology, automatic learning and an optimization algorithm. The performance of the proposed model is compared to baseline approaches.
- *S29 - (Zhao et al., 2021) "Natural language processing-enhanced extraction of SBVR business vocabularies and business rules from UML use case diagrams":* This paper presents a solution as regards the automatic extraction of vocabulary and commercial rules (SBVR). The SBVR model proposed comprises two parts: a commercial vocabulary and commercial rules. The

commercial rules are in the form of a type of glossary that includes substantive concepts and verbal concepts for a specific business domain.

4 Results and Discussion

Before presenting the results of the systematic review, we shall show the results of the quality evaluation, along with their general characteristics.

4.1 Evaluating quality

Evaluating the quality of the selected works is important in order to increase the precision of the data extracted from them. The evaluation determines the validity of the inferences and the credibility and coherence of the summary of the results.

Each of the three researchers scored each work. In the case of disagreements as to the most appropriate score for a particular work, a discussion took place until a consensus was reached. The work was then given an arithmetic means, as shown in Table 3.

The selected papers scored over 42%, with an average of 69.80%, which is considered an acceptable threshold as regards the quality of the studies, as shown in Figure 2.

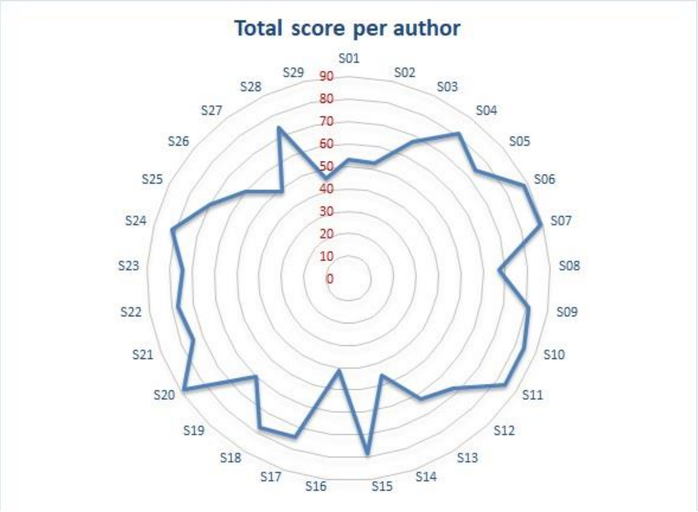

Figure 2: Total score per author

The criteria employed to evaluate the acceptability of the selected papers are based particularly on two categories: 1) artificial intelligence as a mechanism for the resolution of problems in requirements engineering through the use of clearly described techniques and algorithms and the input of ontologies for semantic reasoning, and 2) scenarios that validate the input of the research using tools or case studies and the value that they represent for industry, determining the minimum quality level.

	Author	**Q1**	**Q2**	**Q3**	**Q4**	**Q5**	**Q6**	**Q7**	**Q8**	**Total Score**	**Total %**
S01	Williams. [Williams2018]	1	0.5	1	0.5	0.2	0	0.5	1	4.7	58.80
S02	Hovorushchenko et al. [Hovorushchenkoand Pavlova 2018]	1	0.5	1	0.5	0.2	0	0.5	1	4.7	58.80
S03	Huaqiang et al. [Huaqiang et al. 2017].	1	1	1	0.5	0.2	0	1	1	5.7	71.30
S04	Bargui et al. [Bargui et al. 2011].	1	0.5	1	1	0.2	1	1	1	6.7	83.80
S05	Al-Hroob et al. [Al-Hroob et al. 2018].	1	1	1	1	0.2	1	1	0	6.2	75.50
S06	Di Noia et al. [Di Noia et al. 2018].	1	1	1	1	0.2	1	1	1	7.2	90.00
S07	Wang. [Y. Wang, (2015)].	1	1	1	1	0.2	1	1	1	7.2	90.00
S08	Z. Wang et al. [Z. Wang et al. 2019]	1	0.5	1	1	0.2	0	1	1	6.7	83.80
S09	Coudert et al. [Coudert et al. 2012]	1	0.5	1	1	0.2	1	1	1	6.7	83.80
S10	Morandini et al. [Morandini et al.2017].	0	0.5	1	1	0.4	1	1	1	4.9	61.25
S11	Boukhari et al. [Boukhari et al. 2013]	1	1	1	1	0.4	0.5	1	1	6.9	86.30
S12	Guo, J. Cheng et al. [J. Guo et al. 2017]	1	1	1	1	0.2	0.5	1	0	5.7	71.30
S13	Zait et al. [ZaitandZarour 2018]	1	1	1	0.5	0.4	0.5	1	0	5.4	67.50
S14	Gill et al. [Gill et al. 2014]	1	0.5	1	1	0.2	0	0.5	0	4.2	52.50
S15	Felfernig et al. [Felfernig et al. 2017]	1	0.5	1	1	1	1	1	0	6.5	81.30
S16	Singh et al. [Singh et al. 2016]	1	0.5	1	0.5	0.4	0	0.5	0	3.9	48.80
S17	Hariri et al. [Hariri et al. 2014]	1	1	1	1	0,2	1	1	0	6,2	77.50
S18	Ninaus et al. [Ninaus et al. 2014]	1	1	1	1	0,4	1	1	0	6,4	80.00
S19	K. Annaiahshetty et al. [K. Annaiahshettyand Prasad 2013]	1	1	1	0.5	0.2	1	0.5	0	5.2	65.00
S20	T. Hovorushchenko et al [Hovorushchenko et al. 2019]	1	1	1	1	0.2	1	1	1	7.2	90.00
S21	C.L. Liu [Liu (2016)]	0	0	1	1	0.2	1	1	1	5.2	65.00
S22	O. Emebo et al. [Emebo et al. 2016]	1	0.5	1	1	0.4	1	0.5	1	6.4	80.00
S23	W.-M. Han [Han 2015]	1	1	1	1	0.2	1	1	0	6.2	77.50
S24	J. Guo et al. [Jin Guo et al. 2014b]	1	0.5	1	1	0.2	1	1	1	6.7	83.75
S25	W. Ben AbdessalemKaraa et al. [Ben AbdessalemKaraa et al. 2016]	1	0.5	1	1	0.4	1	1	0	5.9	73.75
S26	MurtazinaandAvdeenko [Murtazina and Avdeenko 2020]	1	0.5	1	0.5	0.2	1	0.5	1	5.7	71.30
S27	Do et al. [Do et al. 2020]	1	0,5	1	1	0,4	0	0,5	0	4,9	61.25
S28	Adithya and Deepak [Adithyaand Deepak, 2021]	1	1	1	1	0.2	0.5	0.5	1	6.2	77.50
S29	Zhao et al. [Zhao et al. 2021]	1	1	1	0.5	0.2	0	0.5	0	4.2	52.20

Table 3: Quality Score

With regard to the first category, which is analyzed in Q1, Q2, Q5 and Q8, it was possible to establish that 93.1% use artificial intelligence techniques, while 74.13% apply artificial intelligence techniques with semantic reasoning using ontologies. However, 41.37% do not employ ontologies.

Despite the fact that two studies obtained a score of 0 for criterion Q1, they were included because we considered that they made an important contribution with the handling of ontologies as part of artificial intelligence.

The second category shows that 82.75% provide input to the community of developers through the creation of academic case studies and prototypes.

4.2 Systematic review results

The results of the systematic review are depicted in Table 4, which summarizes the quantity of studies per initiative, categorized by the application of artificial intelligence.

Type of Initiative	# Studies	Initiative
Expert systems and neural networks	2	S19, S23
Recommendation Systems	3	S06, S15, S17
Case-Based Reasoning (CBR) and Analogy-Based Reasoning (ABR)	2	S09, S22
Natural Language Processing	11	S01, S03, S05, S07, S12, S13, S16, S22, S24, S25, S27
Intelligent Agents	2	S02, S20
Semantic models for the decision-making process	7	S08, S04, S10, S11, S14, S21, S28, S29

Table 4: Summary of quantity of studies by initiative

Table 5, meanwhile, presents the principal contributions in terms of the artificial intelligence methods and techniques used to solve specific problems in requirements engineering. Table 6 shows relevant aspects of the problems and solutions in requirements engineering. Table 7 groups the artificial intelligence technique, the tool used, and the artifact generated, demonstrating where they are present to a greater extent. Finally, Table 8 identifies the requirements engineering phases with the intention of providing a solution by means of intelligent systems.

In Table 4, the intelligent systems have been classified using the cognitive criteria of artificial intelligence, such as: expert systems, decision-support systems, natural interfaces (natural language processing), case-based reasoning (CBR), and intelligent agents. In the present study, these criteria have been considered by classifying and adjusting the initiatives found, which have been differentiated as follows:

Systems based on knowledge (expert systems), which are based on exploiting human knowledge in order to solve problems.

Reasoning based on cases and analogies (CBR), which provide solutions to new problems using the solutions to previous problems as a basis and, in the case of requirements engineering, using already developed projects or software artifacts.

Decision support systems (recommendation systems), as the systems that provide a solution to the problems with the filtering of the information and user preferences and interests.

Natural interfaces (Natural Language Processing), which establish the way in which machines communicate with people using natural language.

Intelligent Agent Systems, which use the information obtained about the environment. This information is then analyzed and compared with the known facts.

	Problem and solution space in RE.	Artificial Intelligent Technique used	Interaction / Similarity calculation	Semantic reasoning (Ontology)	Case study / Tool / Prototype
S01	Obtains security requirements.	NLP	Ontology/actions	General Ontology	Case study (Academy, industry)
S02	Completeness in SRS	Numerical evaluating	SRS /ontology	General Ontology	Case study
S03	Requirement's traceability.	NLP	Origin artifacts/destination artifacts	Not specific	Case study
S04	Analytic requirements in elicitation	Heuristic	Evaluation with domain experts	Specific domain ontology	Tool (ARET)
S05	Use cases from descriptions in Natural Language	NLP Artificial Neural Networks (ANN)	User requirements/ SRS	Not specific	Tool (IT4RE)
S06	RNF and architectural designs	Diffuse logic	Work team/ domain expert solutions	Diffuse ontologies	Case study (academy)
S07	Requirements reuse on (SPL).	Decision trees Stanford Parser	Semantic roles/ERS documents	Domain ontology	Case study (Electronics business)
S08	Requirement's elicitation in the intelligent context of PSS.	Linkage prediction model	Requirements/Intelligent context	General ontology	Case study (smart bicycle design)
S09	Formalizes knowledge for system design	Case-Based Reasoning (CBR)	Requirements (objective) / solutions (sources)	General ontology	Case study
S10	Dynamic changes of requirements	Based on inference rules	Not specific	Not specific	Case study (Academy)
S11	Heterogeneity of vocabulary	Semantic Web Rule Language (SWLR)	Scenario /general ontology	General ontology	Prototype
S12	Traceability requirements	Beep Learning	Among trace evaluation methods	Not specific	Case study (Industrial)
S13	Ambiguity in requirements.	Semantic Web NLP	Similarity among requirements	Not specific	Case study (Academy)
S14	Requirement's ambiguity in open code development community	NLP WSD, Text Mining	Not specific	Not specific	Not specific
S15	Automatic assignment of requirements through reuse	Collaborative Filtering Based recommendation	Project/Users/ Requirements	Not specific	OpenReqTool (Industrial)
S16	Classification of non-functional from SRS.	NLP Rules JAPE	Thematic roles extracted/RNF ISO9126	Not specific	
S17	Analysis of requirements using online forums.	k-nearest neighbors (kNN) incremental diffusive clustering (IDG).	Not specific	Not specific	Case study (academy)
S18	Requirements reuse.	NLP	Input of words/thesaurus	Thesaurus	IntelliReqTool
S19	Representation of experts' knowledge	Based on inference rules	Domain experts/system conclusions	Not specific	Prototype

S20	Metric information in requirements.	Not specific	Requirements /General ontology	General ontology	Case study
S21	RNF and semantic relations of an ontology	Based on inference rules	Not specific	Ontology	Prototype/ Case study (Electronics company)
S22	Requirements implicit	NLP Analogy-Based Reasoning (ABR)	WordNet Ontology /Domain ontology	Domain ontology	IMRTool/ Case study (Academy)
S23	Risks of software projects	Neural Networks (NN)	NN Model/Logistic Regression	Not specific	Case study
S24	Traceability of requirements,	NLP Vector Space Model	Vector Space Model (VSM)/DoCIT	Domain-specific ontology	Prototype
S25	Class Diagram from Natural Language	NLP based on rules, statistics, and patterns.	Not specific	Not specific	Tool (ABCD)
S26	Analysis during agile development	Production rules	Not specific	Domain-ontology	Prototype
S27	New requirements from reuse	Machine learning, NLP	Not specific	Not specific	Case study
S28	Requirement's traceability matrix	Machine learning techniques	Baseline approaches	Domain ontology	Case study
S29	Commercial vocabulary	NLP	UML model/SBVR model	Not specific	Case study

Table 5: Summary of contributions

Table 5 allowed us to create a compendium of some of the selected criteria with the purpose of discovering the input of the initiatives with which to enhance the requirements engineering process using artificial intelligence techniques and semantic reasoning. These criteria included: Problem and space of solution that are solved in Requirements Engineering, technique used in artificial intelligence, interaction or the calculation of similarity, which members of the development team will benefit, the use of ontologies, and there is evidence of the development of tools.

The first criterion in Table 5 was employed to classify and analyze the impact of the problems to be solved, by incorporating artificial intelligence techniques into requirements engineering processes.

Aspects to be solved	**References**
Semantics of the requirements for the definitions and/or structures of SRS	S01, S02, S04, S05, S08, S16, S22
Heterogeneity in the vocabulary	S11, S29
Inconsistency and ambiguity of the requirements	S13, S14
Requirements compared to the analysis and architectonic design of the software and conceptual models	S06, S09, S25, S26
Planning of projects and work teams	S15, S23
Reuse	S07, S18, S27
Management of requirements and traceability	S03, S10, S12, S24, S28

Table 6: Impact on requirements engineering after applying artificial intelligence

This classification is carried out according to the information extracted and how the concepts of natural language are incorporated into requirements engineering processes, independently of whether these processes are traditional or agile. According to the evidence found, the solution space with artificial intelligence helps improve: the semantics in definitions, the heterogeneity of vocabulary, the inconsistence of language, concepts for the construction of architectonic designs, reuse, planning and management. This is illustrated in Table 6.

The technique most frequently used is that of natural language processing (41.37%) for the lexical, syntactic, morphological and semantic analysis of language, which is the input for the generation of new models. The algorithms employed to make decisions are, therefore: heuristics and inference rules (37.9%), neural networks (10.44%), machine learning (6.89%), and Beep learning and the Vector Space Model (3.44%), which make it possible to make inferences about human knowledge.

A total of 4 NLP techniques were found in the studies selected. Table 7 shows the studies which employ: requirements engineering NLP techniques, tools and artifacts.

Reference	NLP technique	Tools	Artifacts
S01	Sentence Parsers	Not specific	Ontology for security requirements
S03	POS & Tokenization	Stanford Parser	Trace Links Semantic
S05	POS	GATE, Stanford-CoreNLP	Use case element classifier (actor, action)
S07	Parsing tree Regular Expressions	Stanford Parser	Semantic Frameworks for Role Labeling
S12	Parsing, Sentiment analysis, Question answering and machine translation	RNNs	Trace Links Semantic
S13	Sentence Parsers	Stanford Parser	Framework to resolve ambiguity
S16	Sentence Parsers	Controlled Natural Language (CNL)	Framework to resolve ambiguity
S22	Part-of-speech (POS) tagging Sentence Parsers	The Apache OpenNLP library wordnet	ToolsImplicit Requirements (IMRs)
S24	Part-of-Speech (POS)	Stanford Parser	Intelligent Traceability Solution (DoCIT)
S25	Sentence Parsers	Stanford Parser	Model UML
S27	POS Tagging Text chunking	Spacy	Framework automated support for innovating requirements
S29	POS	Stanford CoreNLP SimpleNLG	Business vocabularies and business rules

Table 7: NLP techniques, tools and artifacts used in the Selected Studies for Text Processing

According to Table 7, the NLP technique mist frequently used is Part-of- Speech (POS) with the StanfordParser tool as a grammar analyzer. The artifacts generated are frameworks that make it possible to identify key concepts, establish traceability links and resolve ambiguity. The key concepts have been used to generate semantic roles and to assist in the construction of analysis models. However, there is little evidence of studies that use concepts to produce application vocabularies, and that which comes closest is S29.

The calculation of interaction or similarity is, independently of the technique applied, analyzed by evaluating the approximation that the authors give to the solution, on the basis of the given input data compared to the expected outcome, which could be calculated among human experts, artifacts or projects.

The analysis concluded that the greatest number of initiatives calculates the similarity of the solutions of the projects or artifacts previously developed, signifying that they are based on contents or software elements, which provide information for decision making, and only one initiative takes into consideration human wisdom in order to evaluate the solution (S06). In other words, the majority are based on more types of problems in order to evaluate the solution, rather than individual solutions produced by the human intellect.

The application of ontologies as semantic knowledge managers is used in order to make comparisons for the purpose of the disambiguation of words, and these ontologies are applied in order to make decisions regarding definitions and requirements analysis (S01, S03, S07, S11). Meanwhile, S01, S02, S04, S06, S07, S08, S09, S11, S20, S21, S22, S24, S26, S28 use the ontologies to model and represent the knowledge in general or specific domain concepts, allowing inferences and semantic reasoning.

The result of the research works have, above all, been proved and evaluated in the academic field. However, some tools have been evaluated in industry, which uses artificial intelligence to solve problems regarding requirements engineering (S15, S18).

Table 8, which provides a classification of aspects of requirements engineering, such as: activities, types of requirements and those artifacts that could be improved through the use of intelligent systems, shows that the studies have focused on solving problems in the elicitation (30%) and analysis (35%) stages, and that artificial intelligence has been incorporated in order to discover and resolve conflicts among requirements.

This has an impact on elicitation as regards factors such as: the acquisition of domain knowledge, reuse, the preferences and points of view of different stakeholders, the assignation of roles and tasks, and risk analysis during early stages of software projects. The analysis stage impacts on priorization (S09, S16) and disambiguity.

The specification stage (12%) is a consequence of the requirements analysis; however, the contributions in this stage are made as regards recommendations and improvements to the structure of the requirements specification documents.

The use of agile methodologies to provide intelligent support in requirements engineering is shown in one study (S26), and this occurs in the requirements analysis solution area with logical declarations.

The validation stage (5%) attempts to show that the definitions carried out by the user are what the systems should really do. In the literature found, this refers to discovering the quality of requirements by verifying consistency and completeness, which coincides with the analysis because this implies detecting mistakes in the requirements (S18, S15).

The selected studies did not allow us to determine whether the requirements were verified with the users, which could guarantee an agreement between the input context information and the output. A good output would be the fact of including intelligence in these stages by using test cases that can validate what was expected by the user, thus minimizing time and gaps between the elicitation and the tests.

This research has discovered that requirements management is the activity that provides controlled tracking to the four stages of requirements engineering. It has been determined that 18% mention management by means of traceability (S03, S12, S24, S28) and predict dynamic changes (S10).

Initiative	**Requirements engineering stage**					**Types of requirements**		**Requirements engineering artifacts**		
	El	**An**	**Sp**	**Vt**	**Ma**	**Fa**	**Nf**	**AD**	**De**	**MAr**
S01	*						*		*	
S02			*				*		*	
S03					*	*				*
S04	*					*			*	
S05			*			*			*	
S06	*	*					*	*		
S07			*			*	*		*	
S08	*					*			*	
S09		*				*		*		
S10		*			*	*		*		*
S11		*				*		*		
S12					*	*				*
S13									*	
S14		*				*			*	
S15	*	*	*	*	*	*	*		*	
S16		*	*				*		*	
S17	*	*				*			*	
S18		*		*		*		*		
S19	*					*			*	
S20	*	*							*	
S21		*					*			
S22	*				*	*				*
S23	*					*				
S24					*	*				*
S25		*						*		
S26		*				*		*		
S27	*	*			*	*		*		
S28					*	*				*
S29	*					*		*		
%	**30**	**35**	**12**	**5**	**18**	**75**	**25**	**28**	**48**	**24**

Legend:

El = Elicitation An = Analysis Sp = Specification
Va = Validation Ma = Management Fu = Functional
Nf = Non-functional D = Analysis and Design models
De = Definitions and SRS MAr = Management artifacts

Table 8: Application in requirements engineering

Functional requirements (75%) are analyzed to a far greater extent than non-functional requirements in the papers selected. Non-functional requirements (25%) are incorporated with more emphasis on decision making for the system design (S06). Case-Based Reasoning (CBR) is useful for making decisions during design processes.

Artificial intelligence techniques together with non-functional requirements analysis has, in general, been used in order to classify (S16), resolve conflicts (S21) and obtain security requirements (S01).

The software requires both requirements engineering processes and products to be integrated into artificial intelligence environments, and in this respect, the recommendations and decision making for the artifacts produced in requirements engineering are based on requirements definitions and SRS (48%), analysis and design models (28%), contributing to the basic software design architectures and management (24%).

The understanding of the domain is fundamental in the software development process and is the same as that established in the elicitation phase by generating vocabularies. The solution provided in (S29) makes it possible to extract entities using models and by employing the NLP, which allows the generation of design artifacts. We were, however, unable to find studies that define the information in glossaries with semantics as assets for the understanding of the domain, and this is even more important when the production of vocabularies has an impact on the application. We have, therefore, found it necessary to extend the review with a proposal of a methodological process for the construction of a vocabulary that can be applied in specific domains, and which will establish better communication among stakeholders. It will also serve as the starting point for the generation of models on the basis of not only the UML, as determined in the review, but also settings.

5 Proposed Approach

The results obtained in the review made it possible to determine the need to build a new methodological process oriented toward the construction of a vocabulary regarding a specific application through the use of artificial intelligence techniques. This would allow the gap between natural language and the application domain to be filled.

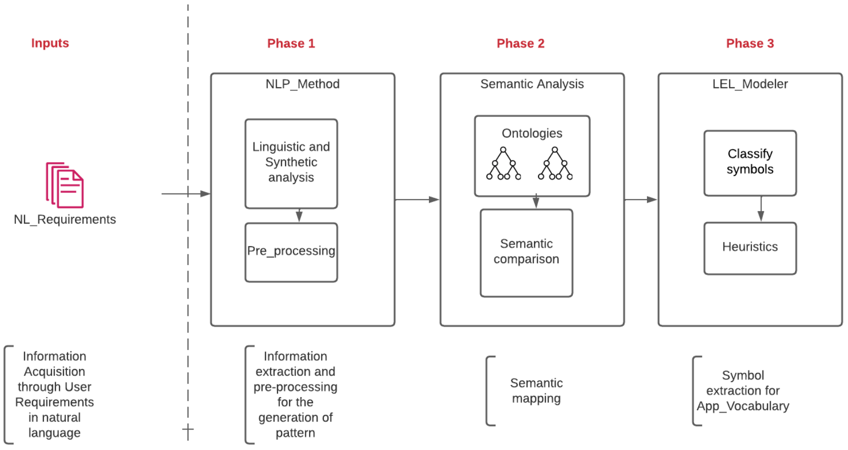
Inputs
Phase 1
Phase 2
Phase 3
NL_Requirements
NLP_Method
Linguistic and Synthetic analysis
Pre_processing
Semantic Analysis
Ontologies
Semantic comparison
LEL_Modeler
Classify symbols
Heuristics
Information Acquisition through User Requirements in natural language
Information extraction and pre-processing for the generation of pattern
Semantic mapping
Symbol extraction for App_Vocabulary

Figure 3: Construction phases of the methodological process

This would make it possible to:

- Generate reference index nouns that would determine the completeness of the vocabulary.
- Analyze discourse actions of the domain by categorizing the verbs, thus making it possible to extend the LEL symbols.
- Determine the semantics of the vocabulary and use relationships to approximate settings, thus providing connotation to the impact of the application.

Figure 3 shows the process employed to build the model, which is divided into three general phases:

- Phase 1, which corresponds to the process of extracting the information regarding the users' requirements, originally produced in natural language, for the linguistic and syntactical analysis by using NLP.
- Phase 2, in which a semantic analysis is presented through the use of ontologies and comparison techniques in order to determine the similarity and concordance among the terms extracted for a specific domain.
- Phase 3, which classifies the symbols in accordance with the Language Extended Lexicon (LEL) in order to form the vocabulary required for the application, using heuristics to determine completeness.

5.1 Phase 1: NLP_Method

The proposed approach is based on the requirements in natural language in order to extract concepts using a linguistic and syntactical analysis so as to then construct patterns that determine the existence of subjects, objects and verbs. A corpus is subsequently used to compare them in order to determine the notion of the vocabulary in the application, as shown in Figure 4.

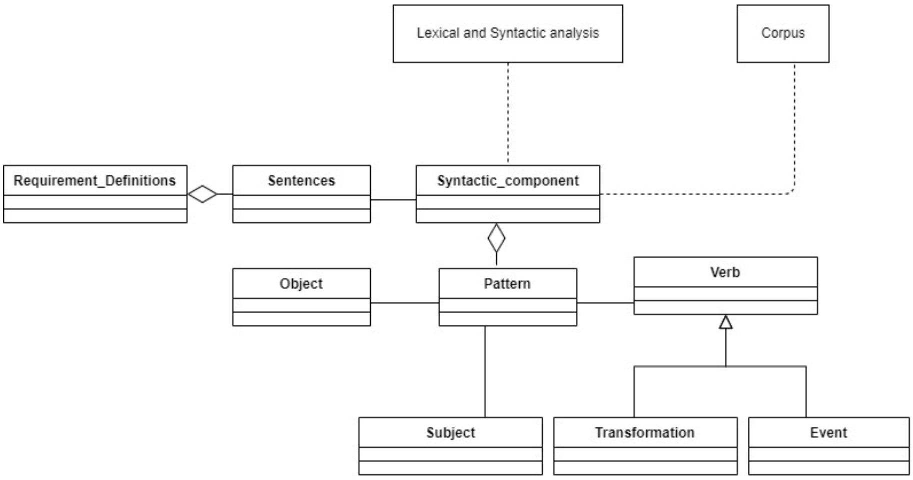

Figure 4: UML model of NLP components for the construction of patterns

The components required for pattern generation are detailed as follows:

- **Sentences:** This is the decomposition of requirements into sentences. The analysis carried out to select sentences is determined by means of the correspondence between a verb and a noun.
- **Syntactic component**: this makes it possible to carry out a linguistic and syntactical analysis of the grammatical structures of the sentences using the following components:
    - Tokenization: This consists of segmenting the text in the sentences selected into fragments that are, in this case, even smaller. However, not all the fragments are useful for analysis, and it is, therefore, necessary to eliminate those words that do not provide value, such as prepositions or adverbs. This objective is attained using Stopwords.
    - POS: This makes it possible to classify (words) into parts of the discourse, such as noun, verb, adjective and pronoun, but not to specify their internal structure or their role in main sentences.
    - SentenceParsers: These are used to identify the elements in the sentence and their grammatical relationships.
- **Pattern**: This generates patterns with which to determine the existence of subjects, objects and verbs as elements of the vocabulary, which are defined as:
    - Subject (Sj): This is defined linguistically as a noun (NN) and semantically as a subject (human users). Example: teacher, student.
    - Object (Ob): This is determined linguistically as being a noun (NN) or as the composition of two nouns ($NN \, \&\& \, NN$). The latter is important as regards showing reference indices that help the completeness of the objects. It is determined semantically as who

receives or is affected by an action. Examples of compound nouns: grade book, enrolment data. This is where the presence of two nouns for each case is clearly visualized.

- Verb (Vb): This is an action, which we have categorized as: 1) the transformation of data (Vbt) – those that do not interact with a subject and which represent an action that is independent of the application. Example: Calculate (Vbt) for the purpose of calculating an operation; and 2) For events (Vbe) – those that directly interact with the subject. Example: Teacher (NN) enters (Vbe) grades in the system (NN_i).

The structure of the relationships that determine the pattern are defined by:

- $RSj(NN, Vb)$ as Subject
- $ROb(NN, Vb, NN) \vee (NN, Vb, NN_i)$ as Object
- $RVbt\ (\rho, Vb, NN_i)\ /\rho = "system"$ as Transformation verb
- $RVbe(\omega, Vb, NN_i)/\omega\ \ = Sj$ as Event verb

An example including the elements explained above is shown below:

1.- Given a noun (NN) and a verb (Vb), the relationship RSj implies the existence of at least one Subject. Example: RSj (teacher, enter).
2.- Given a set of nouns $\{NN_i\ \}$ and a verb (Vb), the relationship ROb implies the existence of at least one object. Example: ROb (teacher, enter, grades). The grades are the only nouns that can denote incompleteness, and it is, therefore, necessary to analyze the existence of a second noun in the structure, such as: ROb (teacher, enter, grades, student). The structure of the predicate considers two nouns in order to show the completeness of the object, as indicated in the structure of ROb.
3.- Given a Verb (Vb) and a set of nouns $\{NN_i\}$, the relationship $RVbt$, in which ρ denotes the action of the system, implies the existence of a transformation verb. Example of $RVbt$(system, calculate, average, grades), as indicated in the structure of $RVbt$.
4.- The cases for the event verb category determine the existence of a subject (ω), and the relationship established for the subjects, therefore, implies the existence of an event verb. Example: $RVbe$ (teacher, enter, grades, student), as indicated in the structure of $RVbe$.

The categorization of these elements creates the basis for the proposed vocabulary.

5.2 Phase 2: Semantic Analysis

The objective of this phase is to take additional information from the syntactic analysis for semantic language processing.

The ontologies make it possible to model the knowledge and represent a “formal and explicit specification of a shared conceptualization”. In order to understand the ontologies, it is vital that they be represented in a concept and relationship space, specifying the meaning and context of a vocabulary for a particular domain. One of the applications of ontologies in Requirements Engineering is that of making it possible to fill the knowledge gap between users and domain experts, which allows their application to this space of the solution. Figure 5 shows the components used, and they are described below:

- **The semantic mapping**: Compares the grammatical cases obtained in Phase 1 with the concepts from the ontology in order to verify whether or not the terms come into conflict with the domain knowledge, establishing the relationship of semantic equivalence. If these elements coincide or are close, they are considered to be part of the application terms, although they are not yet classified
- **Extraction of concepts**: These are the concepts resulting from the systematic mapping and determine the beginning of the minimum vocabulary in an attempt to reduce the use of symbols that are external to the lexicon.
- **Extraction of relationships among concepts**: These are the relationships formed by triplets (subject, predicate, value) that generate the semantics and the context, and which can represent the notion of each symbol while simultaneously allowing the beginning of the construction of scenes.
  The characteristics associated with each binary relationship make it possible to determine whether the concepts are objects or subjects as part of the LEL vocabulary.

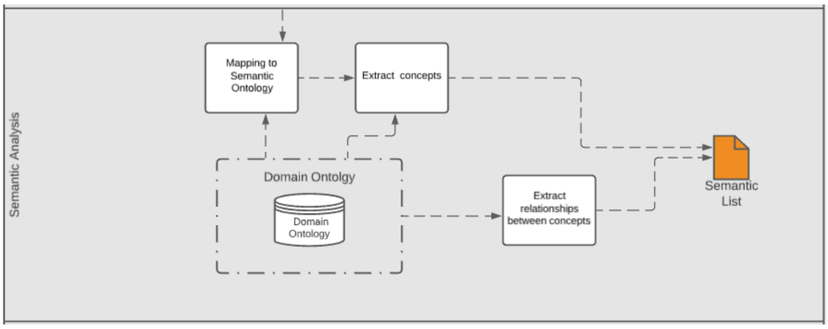

Figure 5: BPMN model of components used to determine semantics

The ontological mapping created with the elements generated and the ontology are: Agent≈subject, action≈verb, recourse≈object, verifying the semantics of the concepts extracted. The ontological concepts are instantiated using the knowledge of the domain, and inferences are established among them. The binary relationship between R(agent, action) semantically establishes the presence of a subject, while the binary relationship R(action, resource) semantically establishes the presence of an object as part of the vocabulary, while simultaneously initiating the construction of settings.

5.3 Phase 3: Building the Vocabulary for the Application

The main purpose of the lexicon is to capture the vocabulary regarding the application domain and its semantics, while the scenes are used to understand the application and its functionality.

The syntax of the LEL is composed of one or more elements, denominated as Symbol. The components of the LEL are represented by the name of the term, the notion and the impact. The name may have synonyms, which are dealt with in the previous phase. The meaning of the term is registered in the notion, while the connotation in the application is registered in the impact.

The vocabulary of the proposed application is based on the validation of the ontology and on the classification of symbols until completeness has been determined.

Figure 6 shows the components employed to construct the vocabulary. Steps for construction:

- **Symbol classification technique**: This constructs rules and heuristics with which to classify according to the type of symbol: Subject, object, verb or state.
- **Classified Symbols**: If a symbol is related to one or various sentences from the semantic list, it is considered to be part of the vocabulary and has a notion and an impact. Otherwise, it is not classified. New terms can appear in the notion and impact, thus establishing the beginning of circularity and enabling them to become valid again.
- **Non-classified symbols**: These are symbols and are established by means of a manual process. This requires the intervention of the domain expert to verify whether or not they are part of the vocabulary, and if they are, the ontology is again fed in order to generate new knowledge.

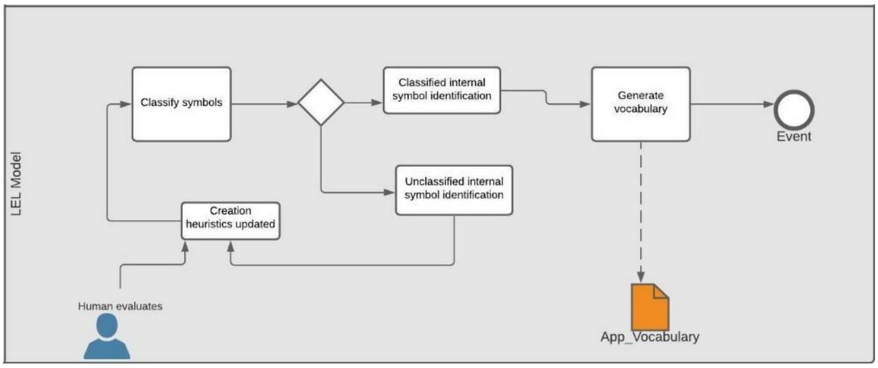
LEL Model
Classify symbols
Classified internal symbol identification
Generate vocabulary
Event
Unclassified internal symbol identification
Creation heuristics updated
Human evaluates
App_Vocabulary

Figure 6: BPMN model of components for the construction of the vocabulary

The methodological process in which all the components are integrated and related is shown in Figure 7.

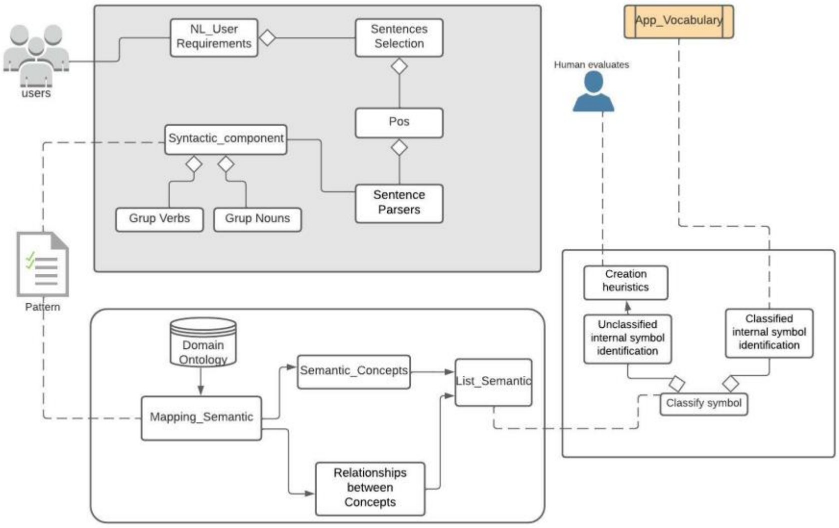

Figure 7: Methodological process for the Vocabulary of the application

6 Current state and threats to validity

The objective of the proposed approach is to evaluate the requirements engineering process on the basis of Intelligent Systems and Semantic Reasoning using manual and automatic sub-processes. The manual part will be carried out by experts in this domain, who will employ heuristics to classify the terms related to the vocabulary. The model also seeks to automate the extraction of information in order to calculate the precision of the requirements collection. This will be done by applying ontologies and machine learning techniques.

The model is currently being validated in the education domain. Despite being at an early stage, it has been possible to identify some of the benefits of the proposal, such as helping describe the requirements in natural language, thus allowing a better classification of information. The application of the rules and principles of the language enables the language to be processed, and a complete grammar to be obtained.

For example, in the sentence "The grades are registered in the system", the domain expert can classify the term "grades" as a vocabulary input by explicitly determining the Object symbol. The complete grammatical structure should be: "the professor logs the student's grades in the system", where "professor" is the Subject and "the student's grades" is the Object.

In the example above, the incompleteness is given by the nouns for which it is necessary to create reference indices. The actions can similarly be classified by employing verbs through the use of patterns, as described in subsection 5.1.

Finally, the semantics and consistency of the sentences in a determined domain and context is evaluated by applying the ontology. This ontology is, in turn, supported by machine learning techniques, which make it possible to determine a value for the precision of the extraction.

On the other hand, the main risk faced with the proposal, is that, the raising of a question leads to the analysis of a single factor concerning a topic, and there may be bias in the selection of studies or the data may not be extracted correctly.

We have attempted to deal with this threat by defining tables containing analysis factors for data extraction that cover the research question. We have identified key words and search terms that have allowed us to identify relevant studies, including quality criteria with which to evaluate the credibility of the studies. It is not, however, possible to guarantee that all the studies that could have answered the research question were identified.

Systematic errors may occur when designing how the review is to be carried out. We attempted to avoid this threat by selecting a review protocol based on methods, but it was, in turn, necessary to simplify certain sections of the review and illustrate them using figures.

Biases in the review could be avoided by extending the research group.

7 Conclusions

The objective of this paper was to employ a conceptual framework in order to discover those research works that have explored the use of intelligent support in requirements engineering activities by employing the knowledge contained in ontologies.

In this review, we have analyzed and identified five types of intelligent systems with cognitive criteria regarding artificial intelligence, which support RE processes.
According to the literature researched, there are more recommendation systems than expert systems.

The interaction of intelligent systems with ontologies for the resolution of problems has been driven by work diagrams, in which the ontologies for some research have, during the definitions of requirements, served as syntactic and semantic comparators for grammatical structures tagged from artifacts for the management and the acquisition of knowledge from specific domains. However, the total incorporation of semantics through the use of requirements ontologies has yet to be achieved.

The most frequently used technique is natural language processing (NLP), which is used to complement other artificial intelligence techniques in order to provide intelligent support to the requirements, since it is useful as regards reducing ambiguity, inconsistency and/or incompleteness.

The artificial intelligence techniques such as (CBR) Case Based Reasoning found in the literature studied are applied in order to search for solutions regarding the system design, using non-functional requirements as a basis on which to define the architectures.

The review process carried out found few results regarding the extraction of vocabularies for application to specific domains by incorporating intelligence and semantics in the initial stages of requirements engineering. This should be independent of the method used by the development team, thus making it possible to reduce the time and effort required to construct software artifacts. A solution is, therefore, provided by means of the proposed approach.

The contribution of this research is that of providing researchers with a summary of the existing information regarding Intelligent Systems in requirements engineering in an exhaustive and impartial manner, in order to provide a context in which they can work on requirements engineering activities and that have been the support for the software development team.

This research is, therefore, a comparison of the opinions that exist at present, whilst ensuring the reliability of the information and of the results obtained.

The subject of ontologies as work diagrams for intelligent systems is still new, and the planning of systematic reviews is limited. This study may, therefore, be distorted owing to the fact that there is little information in these fields.

Acknowledgements

This work has been co-funded by the GENESIS project (Consejería de Educación, Cultura y Deportes de la Junta de Comunidades de Castilla La Mancha, y Fondo Europeo de Desarrollo Regional FEDER, SBPLY/17/180501/000202), and the AETHER-UCLM proyect (MCIN/AEI/10.13039/501100011033, PID2020-112540RB-C42).

References

[Afzal et al., 2016] Afzal, U., Mahmood, T., and Shaikh, Z. (2016). Intelligent software productline configurations: A literature review. Computer Standards and Interfaces, 48, 30-48.

[Adithya et al., 2021] Adithya, V., and Deepak, G. (2021). OntoReq: An Ontology Focused Collective Knowledge Approach for Requirement Traceability Modelling. European, Asian, Middle Eastern, North African Conference on Management and Information Systems, 358-370.

[Al-Hroob et al., 2018] Al-Hroob, A., Imam, A. T., and Al-Heisa, R. (2018). The use of artificial neural networks for extracting actions and actors from requirements document. Information and Software Technology, 101, 1-15.https://doi.org/10.1016/j.infsof.2018.04.010

[AlZu'bi et al., 2018] AlZu'bi, S., Hawashin, B., EIBes, M., and Al-Ayyoub, M. (2018). A novel recommender system based on apriori algorithm for requirements engineering. 2018 fifth international conference on social networks analysis, management and security (snams), 323-327.https://doi.org/10.1109/SNAMS.2018.8554909

[Andrews et al., 2014] Andrews, Z., Ingram, C., Payne, R., Romanovsky, A., Holt, J., and Perry, S. (2014). 3.2. 2 Traceable Engineering of Fault-Tolerant SoSs. INCOSE International Symposium, 24, 258-273.https://doi.org/10.1002/j.2334-5837.2014.tb03148.x

[Annaiahshetty and Prasad, 2013] Annaiahshetty, K., and Prasad, N. (2013). Expert System for Multiple Domain Experts Knowledge Acquisition in Software Design and Development. 2013 UKSim 15th International Conference on Computer Modelling and Simulation, 196-201. https://doi.org/10.1109/UKSim.2013.124

[Annaiahshetty et al., 2013] Annaiahshetty, Kalashankar, and Prasad, N. (2013). Expert System for Multiple Domain Experts Knowledge Acquisition in Software Design and Development. Computer Modelling and Simulation (UKSim), 2013 UKSim 15th International Conference on, 196-201.https://doi.org/10.1109/UKSim.2013.124

[Bargui et al., 2011] Bargui, F., Ben-Abdallah, H., and Feki, J. (2011). A decision making ontology building process for analytical requirements elicitation. 2011IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications, 1529-1536. https://doi.org/10.1109/TrustCom.2011.212

[Basirati et al., 2020] Basirati, M. R., Otasevic, M., Rajavi, K., Böhm, M., andKrcmar, H. (2020).Understanding the relationship of conflict and success in software development projects. Information and Software Technology, 126, 106331. https://doi.org/10.1016/j.infsof.2020.106331

[Ben et al., 2016] Ben AbdessalemKaraa, W., Ben Azzouz, Z., Singh, A., Dey, N., S. Ashour, A., and Ben Ghazala, H. (2016). Automatic builder of class diagram (ABCD): An application of UML generation from functional requirements. Software: Practice and Experience, 46(11), 1443-1458. https://doi.org/10.1002/spe.2384

[Bhatia et al., 2016] Bhatia, M. P. S., Kumar, A., and Beniwal, R. (2016). Ontologies for software engineering: Past, present and future. Indian Journal of Science and Technology, 9(9).

[Binkhonain, M and Zhao, L,2019] Binkhonain, M., and Zhao, L (2019). A review of machine learning algorithms for identification and classification of non-functional requirements. Expert Systems with Applications: X, 1, 100001.https://doi.org/10.1016/j.eswax.2019.100001.

[Boukhari et al., 2013] Boukhari, I., Bellatreche, L., and Khouri, S. (2013). Efficient, Unified, and Intelligent User Requirement Collection and Analysis in Global Enterprises. Proceedings of International Conference on Information Integration and Web-Based Applications and Services - IIWAS '13, 686-691. https://doi.org/10.1145/2539150.2539263

[Bouquet et al., 2003] Bouquet, P., Ghidini, C., Giunchiglia, F., and Blanzieri, E. (2003). Theories and uses of context in knowledge representation and reasoning. Journal of pragmatics, 35(3), 455-484.https://doi.org/10.1016/S0378-2166(02)00145-5

[Bozyiğit et al., 2021] Bozyiğit, F., Aktaş, Ö., andKılınç, D. (2021). Linking software requirements and conceptual models: A systematic literature review. Engineering Science and Technology, an International Journal, 24(1), 71-82.https://doi.org/10.1016/j.jestch.2020.11.006

[Brereton et al., 2008] Brereton, P., Kitchenham, B. A., Budgen, D., and Li, Z. (2008). Using a Protocol Template for Case Study Planning. EASE, 8, 41-48.https://doi.org/10.14236/ewic/EASE2008.5

[Clark et al., 2021] Clark, A., Walkinshaw, N., and Hierons, R. M. (2021). Test case generation for agent-based models: A systematic literature review. Information and Software Technology, 106567.https://doi.org/10.1016/j.infsof.2021.106567

[Coudert et al., 2012] Coudert, T., Vareilles, E., Geneste, L., and Aldanondo, M. (2012). Improvement of retrieval in case-based reasoning for system design. 2012 IEEE International Conference on Industrial Engineering and Engineering Management, 1538-1542. https://doi.org/10.1109/IEEM.2012.6838004

[Curcio et al., 2018] Curcio, K., Navarro, T., Malucelli, A., and Reinehr, S. (2018). Requirements engineering: A systematic mapping study in agile software development. Journal of Systems and Software, 139, 32-50.https://doi.org/10.1016/j.jss.2018.01.036

[Dermeval et al., 2016] Dermeval, D., Vilela, J., Bittencourt, I. I., Castro, J., Isotani, S., Brito, P., and Silva, A. (2016). Applications of ontologies in requirements engineering: A systematic review of the literature. Requirements Engineering, 21(4), 405-437. https://doi.org/10.1007/s00766-015-0222-6

[Di Noia et al., 2018] Di Noia, T., Mongiello, M., Nocera, F., and Straccia, U. (2018). A fuzzy ontology-based approach for tool-supported decision making in architectural design. Knowledge and Information Systems, 1-30. https://doi.org/10.1007/s10115-018-1182-1

[Dybå, T., and Dingsøyr, T. 2008] Dybå, T., and Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. Information and software technology, 50(9-10), 833-859.https://doi.org/10.1016/j.infsof.2008.01.006

[Dick et al., 2017] Dick, J., Hull, E., and Jackson, K. (2017). Requirements Engineering. Springer. https://doi.org/10.1007/978-3-319-61073-3_3

[do Prado Leite et al., 2000] do Prado Leite, J. C. S., Hadad, G. D., Doorn, J. H., and Kaplan, G. N. (2000). A scenario construction process. Requirements Engineering, 5(1), 38-61.https://doi.org/10.1007/PL00010342

[Emebo et al., 2016] Emebo, O., Olawande, D., and Charles, A. (2016). An automated tool support for managing implicit requirements using Analogy-based Reasoning. 2016 IEEE Tenth International Conference on Research Challenges in Information Science (RCIS), 1-6. https://doi.org/10.1109/RCIS.2016.7549329

[Felfernig et al., 2017] Felfernig, A., Stetinger, M., Falkner, A., Atas, M., Franch Gutiérrez, J., and PalomaresBonache, C. (2017). OpenReq: Recommender systems in requirements engineering. Proceedings of the Workshop Papers of i-Know 2017: co-located with International Conference on Knowledge Technologies and Data-Driven Business 2017 (i-Know 2017): Graz, Austria, October 11-12, 2017, 1-4.http://hdl.handle.net/2117/119016

[Gayathri and Uma, 2018] Gayathri, R., and Uma, V. (2018). Ontology based knowledge representation technique, domain modeling languages and planners for robotic path planning: A survey. ICT Express, 4(2), 69-74.https://doi.org/10.1016/j.icte.2018.04.008

[Gill et al., 2014] Gill, K. D., Raza, A., Zaidi, A. M., andKiani, M. M. (2014). Semi-automation for ambiguity resolution in Open Source Software requirements. 2014 IEEE 27th Canadian Conference on Electrical and Computer Engineering (CCECE), 1-6. https://doi.org/10.1109/CCECE.2014.6900955

[Guo et al., 2017] Guo, J., Cheng, J., and Cleland-Huang, J. (2017). Semantically Enhanced Software Traceability Using Deep Learning Techniques. 2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE), 3-14. https://doi.org/10.1109/ICSE.2017.9

[Guo et al., 2014] Guo, Jin, Monaikul, N., Plepel, C., and Cleland-Huang, J. (2014a). Towards an intelligent domain-specific traceability solution. Proceedings of the 29th ACM/IEEE international conference on Automated software engineering, 755-766.https://doi.org/10.1145/2642937.2642970

[Guo et al., 2014b] Guo, Jin, Monaikul, N., Plepel, C., and Cleland-Huang, J. (2014b). Towards an intelligent domain-specific traceability solution. Proceedings of the 29th ACM/IEEE international conference on Automated software engineering, 755-766.https://doi.org/10.1145/2642937.2642970

[Gupta and Nagpal, 2020] Gupta, I., and Nagpal, G. (2020). Artificial Intelligence and Expert Systems. Stylus Publishing, LLC

[Han, 2015] Han, W.-M. (2015). Discriminating risky software project using neural networks. Computer Standards and Interfaces, 40, 15-22.https://doi.org/10.1016/j.csi.2015.01.001Get rights and content

[Hariri et al., 2014] Hariri, N., Castro-Herrera, C., Cleland-Huang, J., and Mobasher, B. (2014). Recommendation Systems in Requirements Discovery. En M. P. Robillard, W. Maalej, R. J. Walker, and T. Zimmermann (Eds.), Recommendation Systems in Software Engineering (pp. 455-476). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-45135-5_17

[Hovorushchenko and Pavlova, 2018] Hovorushchenko, T., and Pavlova, O. (2018). Evaluating the Software Requirements Specifications Using Ontology-Based Intelligent Agent. 2018 IEEE 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT), 1, 215-218. https://doi.org/10.1109/STC-CSIT.2018.8526730

[Huaqiang et al., 2017] Huaqiang, D., Hongxing, L., Songyu, X., and Yuqing, F. (2017). The research of domain ontology recommendation method with its applications in requirement traceability. 2017 16th International Symposium on Distributed Computing and Applications to Business, Engineering and Science (DCABES), 158-161. https://doi.org/10.1109/DCABES.2017.42

[Jennings, 2000] Jennings, N. R. (2000). On agent-based software engineering. Artificial intelligence, 117(2), 277-296.

[Jureta et al.,] Jureta, I., Mylopoulos, J., and Faulkner, S. (2008). Revisiting the core ontology and problem in requirements engineering. 2008 16th IEEE International Requirements Engineering Conference, 71-80. http://doi.org/10.1109/RE.2008.13

[Kaur et al., 2021] Kaur, K., Singh, P., and Kaur, P. (2021). A Review of Artificial Intelligence Techniques for Requirement Engineering. Computational Methods and Data Engineering, 259-278.

[Kitchenham and Brereton, 2013] Kitchenham, B., and Brereton, P. (2013). A systematic review of systematic review process research in software engineering. Information and software technology, 55(12), 2049-2075

[Landhäußer et al., 2014] Landhäußer, M., Körner, S. J., and Tichy, W. F. (2014). From requirements to UML models and back: How automatic processing of text can support requirements engineering. Software Quality Journal, 22(1), 121-149.https://doi.org/10.1007/s11219-013-9210-6

[Laplante, 2017] Laplante, P. A. (2017). Requirements engineering for software and systems. Auerbach

[Lee et al., 2016] Lee, Y.-C., Eastman, C. M., andSolihin, W. (2016). An ontology-based approach for developing data exchange requirements and model views of building information modeling. Advanced Engineering Informatics, 30(3), 354-367. https://doi.org/10.1016/j.aei.2016.04.008

[Maalej et al., 2015] Maalej, W., Nayebi, M., Johann, T., and Ruhe, G. (2015). Toward data-driven requirements engineering. IEEE Software, 33(1), 48-54. https//doi.org/10.1109/MS.2015.153

[Morandini et al., 2017] Morandini, M., Penserini, L., Perini, A., and Marchetto, A. (2017). Engineering requirements for adaptive systems. Requirements Engineering, 22(1), 77-103. https://doi.org/10.1007/s00766-015-0236-0

[Murtazina and Avdeenko, 2020] Murtazina, M., and Avdeenko, T. (2020). The ontology-driven approach to intelligent support of requirements engineering in agile software development. 2020 International Conference on Information Technology and Nanotechnology (ITNT), 1-6. https://doi.org/10.1109/ITNT49337.2020.9253232

[Ninaus et al., 2014] Ninaus, G., Felfernig, A., Stettinger, M., Reiterer, S., Leitner, G., Weninger, L., and Schanil, W. (s. f.). INTELLIREQ: Intelligent Techniques for Software Requirements Engineering. 6.

[Ninaus et al., 2014] Ninaus, G., Reinfrank, F., Stettinger, M., and Felfernig, A. (2014). Content-based recommendation techniques for requirements engineering. Artificial Intelligence for Requirements Engineering (AIRE), 2014 IEEE 1st International Workshop on, 27-34. https://doi.org/10.1109/AIRE.2014.6894853

[Nunes et al., 2020] Nunes, I., Treude, C., andCalefato, F. (2020). The Impact of Dynamics of Collaborative Software Engineering on Introverts: A Study Protocol. Proceedings of the 17th International Conference on Mining Software Repositories, 619-622. https://doi.org/10.1145/3379597.3387505

[Pakdeetrakulwong et al., 2013] Pakdeetrakulwong, Udsanee, and Wongthongtham, P. (2013). State of the art of a multi-agent based recommender system for active software engineering ontology. International Journal of Digital Information and Wireless Communications (IJDIWC), 3(4), 363-376.

[Perkusich et al., 2020] Perkusich, M., e Silva, L. C., Costa, A., Ramos, F., Saraiva, R., Freire, A., Dilorenzo, E., Dantas, E., Santos, D., andGorgônio, K. (2020). Intelligent software engineering in the context of agile software development: A systematic literature review. Information and Software Technology, 119, 106241. https://doi.org/10.1016/j.infsof.2019.106241

[Preece, 2018] Preece, A. (2018). Asking 'Why' in AI: Explainability of intelligent systems – perspectives and challenges. Intelligent Systems in Accounting, Finance and Management, 25(2), 63-72. https://doi.org/10.1002/isaf.1422

[Salado and Tan, 2020] Salado, A., and Tan, R. M. (2020). Structural Rules for an Intelligent Advisor to Identify Requirements Gaps using Model-Based Requirements. 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 920-927. https://doi.org/10.1109/SMC42975.2020.9283466

[Singh, 2016] Singh, P., Singh, D., and Sharma, A. (2016). Rule-based system for automated classification of non-functional requirements from requirement specifications. 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 620-626. https://doi.org/10.1109/ICACCI.2016.7732115

[Urbieta et al., 2020] Urbieta, M., Antonelli, L., Rossi, G., & do Prado Leite, J. C. S. (2020). The impact of using a domain language for an agile requirements management. Information and Software Technology, 127, 106375.

[Vasanthapriyan et al., 2015] Vasanthapriyan, S., Tian, J., and Xiang, J. (2015). A survey on knowledge management in software engineering. 2015 IEEE International Conference on Software Quality, Reliability and Security-Companion, 237-244. https://doi.org/10.1109/QRS-C.2015.48

[Wang, 2015] Wang, Y. (2015). Semantic information extraction for software requirements using semantic role labeling. 2015 IEEE International Conference on Progress in Informatics and Computing (PIC), 332-337. https://doi.org/10.1109/PIC.2015.7489864

[Wang et al., 2019] Wang, Z., Chen, C.-H., Zheng, P., Li, X., and Khoo, L. P. (2019). A novel data-driven graph-based requirement elicitation framework in the smart product-service system context. Advanced engineering informatics, 42, 100983.https://doi.org/10.1016/j.aei.2019.100983

[Williams, 2018] Williams, I. (2018). An Ontology Based Collaborative Recommender System for Security Requirements Elicitation. 2018 IEEE 26th International Requirements Engineering Conference (RE), 448-453. https://doi.org/10.1109/RE.2018.00060

[Zait and Zarour, 2018] Zait, F., and Zarour, N. (2018). Addressing Lexical and Semantic Ambiguity in Natural Language Requirements. 2018 Fifth International Symposium on Innovation in Information and Communication Technology (ISIICT), 1-7. https://doi.org/10.1109/ISIICT.2018.8613726

[Zhao et al., 2021] Zhao, L., Alhoshan, W., Ferrari, A., Letsholo, K. J., Ajagbe, M. A., Chioasca, E.-V., and Batista-Navarro, R. T. (2021). Natural Language Processing for Requirements Engineering: A Systematic Mapping Study. ACM Computing Surveys (CSUR), 54(3), 1-41.https://doi.org/10.1145/3444689