IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 16, 2022, accepted June 7, 2022, date of publication June 13, 2022, date of current version June 17, 2022.
*Digital Object Identifier 10.1109/ACCESS.2022.3182372*

# The Use of NLP-Based Text Representation Techniques to Support Requirement Engineering Tasks: A Systematic Mapping Review

**RIAD SONBOL[1], GHAIDA REBDAWI[1], AND NADA GHNEIM[2]**
[1]Department of Informatics, Higher Institute for Applied Sciences and Technology (HIAST), Damascus, Syria
[2]Faculty of Information and Communication Technology, Arab International University (AIU), Damascus, Syria
Corresponding author: Riad Sonbol (riad.sonbol@hiast.edu.sy)

**ABSTRACT** Natural Language Processing (NLP) is widely used to support the automation of different Requirements Engineering (RE) tasks. Most of the proposed approaches start with various NLP steps that analyze requirements statements, extract their linguistic information, and convert them to easy-to-process representations, such as lists of features or embedding-based vector representations. These NLP-based representations are usually used at a later stage as inputs for machine learning techniques or rule-based methods. Thus, requirements representations play a major role in determining the accuracy of different approaches. In this paper, we conducted a survey in the form of a systematic literature mapping (classification) to find out (1) what are the representations used in RE tasks literature, (2) what is the main focus of these works, (3) what are the main research directions in this domain, and (4) what are the gaps and potential future directions. After compiling an initial pool of 2,227 papers, and applying a set of inclusion/exclusion criteria, we obtained a final pool containing 104 relevant papers. Our survey shows that the research direction has changed from the use of lexical and syntactic features to the use of advanced embedding techniques, especially in the last two years. Using advanced embedding representations has proved its effectiveness in most RE tasks (such as requirement analysis, extracting requirements from reviews and forums, and semantic-level quality tasks). However, representations that are based on lexical and syntactic features are still more appropriate for other RE tasks (such as modeling and syntax-level quality tasks) since they provide the required information for the rules and regular expressions used when handling these tasks. In addition, we identify four gaps in the existing literature, why they matter, and how future research can begin to address them.

**INDEX TERMS** Natural language processing, requirements engineering, requirements representation, syntax, semantic.

## I. INTRODUCTION

Requirements Engineering (RE) is the most critical phase of the software development life cycle [1]. It aims to specify precisely the requirements that must be met or possessed by the desired system [2]. RE involves a wide range of tasks related to extracting, documenting, analyzing, validating, and managing requirements [3]. The requirement represents the core elements of all these tasks. It is a broad concept describing a purpose, a need, a goal, a functionality, a constraint, a quality, a behavior, a service, a condition, or a capability [4]. Requirements play a major role in the success or failure of software projects [5]. According to many studies, most errors in software development projects stem from the RE phase. The lack of understanding of requirements increases the risk of time and cost overrun of the project [3], [5].

Requirements come from various stakeholders who have different needs, roles, and responsibilities, and are as such prone to the occurrence of conflicts including interference, interdependency, and inconsistency [6]. Moreover, requirements typically are specified in natural languages, which increases the complexity of requirement engineering because of the inherent ambiguity, incompleteness, and inaccuracy of natural languages [7]. These factors make RE tasks challenging, time-consuming, and error-prone work mainly for large projects, as large volumes of requirements need to be processed, analyzed, and understood [8].

Many researches have been carried out on the automation of different RE tasks. The proposed approaches usually start by applying a set of Natural Language Processing (NLP)

The associate editor coordinating the review of this manuscript and approving it for publication was Arianna Dulizia.

steps that extract linguistic features and information from requirements texts and construct various NLP-based representations. These representations are used in further stages to solve the targeted task (e.g., classifying requirements [9]–[11], detecting trace links, discovering quality defects, etc.).

In the last decade, a growing number of works employed various syntactic and semantic-based features to represent requirements. The used representation is considered one of the most important factors that affect the accuracy of the proposed solutions for different RE tasks [12], [13]. Therefore, there is a need to analyze the related literature to investigate the suggested representations, identify gaps, and guide future research.

To address the aforementioned needs, a systematic mapping review was used as a research method for this study. A mapping review is aimed at providing an understanding of the scope of the research activities in a given area [14]. Compared to a traditional literature review, a mapping review has many advantages, such as a well-defined methodology that reduces bias and a wider context that allows more general conclusions [15]. The review presented in this paper aims to identify all recently published primary literature that employs NLP-based representations to support RE tasks. We classify these works based on two aspects: the targeted RE task and the proposed representation, and identify potentials and gaps in the field to inform future research.

The rest of the paper is structured as follows: In section 2 we provide a background for NLP-related concepts. Section 3 provides a quick overview of related works. Section 4 describes the methodology adopted to conduct this study including the search terms, online databases, and the systematic mapping process. Section 5 presents and discusses our results. We summarize our findings in section 6 and conclude our paper in section 7.

## II. BACKGROUND: NATURAL LANGUAGE PROCESSING

Natural language processing is one of the main artificial intelligence disciplines. It aims to enable computer programs to "understand" and process natural language texts to achieve some specific goals [16], [17]. Traditionally, there are three main levels of processing in an NLP-based approach: [18]: lexical and morphological level, syntactic level, and semantic level.

### A. MORPHOLOGICAL (OR LEXICAL) LEVEL

The Morphological level focuses on analyzing words into their morphemes like prefixes, suffixes, and base words. It includes common tasks, such as Tokenization and Lemmatization [18].

#### 1) TOKENIZATION

the process of splitting a text into a list of tokens. Tokens can be words, numbers, or punctuation marks.

#### 2) LEMMATIZATION

the process of finding the dictionary form, or the lemma, of each word. For example, the lemma of *"Supporting"* and *"Supported"* is *"Support"*.

### B. SYNTACTIC LEVEL

The Syntactic level focuses on analyzing the grammatical structure of sentences. This level usually includes Part-of-Speech Tagging (POS-tagging), Chunking, dependency Parsing, and Named-Entity Recognition [18].

#### 1) POS-TAGGING

the process of tagging each token in a sentence with its corresponding part of a speech tag (such as noun, verb, adjective, etc.) based on its syntactical context [19].

#### 2) CHUNKING

the process of detecting syntactic constituents such as Noun Phrases and Verb Phrases in a sentence [20].

#### 3) DEPENDENCY PARSING

the process of analyzing the syntactic structure of the sentence, by finding out the grammatically related words, as well as the type of the relationship between them [21].

#### 4) NAMED-ENTITY RECOGNITION

It seeks to locate and classify named entities mentioned in the sentence into pre-defined categories such as person names, organizations, locations, etc [22].

### C. SEMANTIC LEVEL

The Semantic level focuses on understanding the meaning of the text. The main goal of semantic processing is to automatically map a natural language sentence into a formal representation of its meaning. Different semantic representations have been proposed in the literature, such as:

#### 1) ONTOLOGY-BASED REPRESENTATION

The ontology is a data model that represents a set of concepts within a domain and the relationships between those concepts [23]. WordNet [24] is one of the widely used lexical ontologies. Ontologies are commonly used to assign words to a predefined set of concepts and to measure semantic similarity between them using different ontology-based measures such as Wu and Palmer [25] and path-based similarity [26].

#### 2) VECTOR SPACE MODEL (VSM)

A basic representation model that represents text as a term-by-document matrix [27]. The Bag-of-Words (BOW) model is a special case for VSM where words frequencies are used as weights, and words are used as features. Other weighting factors are used in VSM, such as IDF and TF-IDF.

#### 3) TOPIC MODELING-BASED REPRESENTATION

A statistical modeling approach used to discover the latent or abstract topics that occur in a set of texts. These topics are used to represent each text. This approach helps in finding a low-rank approximation to the term-document matrix by retaining the semantic relations between words. Latent Dirichlet Allocation (LDA) [28] and Latent Semantic Analysis (LSA) [29] are two common examples of this approach.

#### 4) ADVANCED EMBEDDING TECHNIQUES

Embedding is an efficient method for learning high-quality vector representations of words from large amounts of unstructured text data [30]. Word embedding can capture the context of a word within a document, which allows words with similar meanings to have similar vector representations. Many famous pre-trained word embeddings are available to the public, such as Word2Vec [30], GloVe [31], BERT [32], etc.

## III. RELATED REVIEWS

Many reviews have been published on the relation between NLP and RE tasks [33]–[35].

Some of these reviews provide a broad picture of NLP activities and technologies used in the RE domain [33], [36], [37]. Dermeval *et al.* [36] conducted a systematic literature review to identify the primary studies on the use of ontologies in the RE domain. This study considered 77 studies published between January 2007 and October 2013. Nazir *et al.* [37] investigated the applications of NLP in the context of RE. It included 27 studies published between 2002 and 2016. Zhao *et al.* [33] introduced a comprehensive overview of the applications of NLP in RE research focusing on the state of the literature, the state of empirical research, the research focus, the state of the practice, and the NLP technologies used. This study reviewed 404 relevant primary studies reported between 1983 and April 2019.

Other reviews focused on specific RE problems. Bozyigit *et al.* [38] provided a review of 44 primary studies related to the automatic transformation of software requirements into conceptual models. It covered works published between 1996 and 2020.

On the other hand, a number of reviews have limited their work to specific templates for requirements [34], [35]. Amna *et al.* [35] reviewed studies that investigate or develop solutions for problems related to ambiguity in user stories. The study covered 36 researches published between 2001 and 2020. Similarly, Raharjana *et al.* [34] presented a systematic literature review for research related to the role of NLP on user story specification. This work found 30 primary studies between January 2009 to December 2020.

Although all these works provided good information regarding requirements engineering, no conducted secondary studies have focused on the used techniques to represent textual requirements and the involved syntactic and semantic aspects in these representations.

## IV. RESEARCH METHOD

This study was undertaken as a systematic mapping review using the guidelines presented in Petersen *et al.* [15]. The goal of this review is to identify, categorize, and analyze existing literature published between 2010 and 2021 and use syntactic and semantics aspects to represent software requirements when handling RE tasks.

### A. PLANNING

In this section, we define our research questions, the search strategy we use, and the inclusion and exclusion criteria considered to filter the results.

#### 1) RESEARCH QUESTION

Our work is guided by the following main research questions:

*"How are the syntactic and semantics aspects considered to represent software requirements when handling RE tasks?"*.

This question is detailed in the following five RQs:

- RQ1: What is the state of the published literature on RE works that use syntactic and semantic representations for requirements?
- RQ2: In which RE tasks are the syntactic and semantic aspects mostly considered to represent requirements?
- RQ3: What are the proposed representations in the literature for RE tasks?
- RQ4: What are the main research directions to represent requirements for each category of RE tasks?
- RQ5: What gaps and potential future directions exist in this field?

While the first question (RQ1) focuses on having a general overview of the published works (number of publications and the top publication venues), the second and the third questions analyze the targeted problems in RE (RQ2) and the proposed solutions (RQ3). The fourth research question (RQ4) explores the current research directions for each category of RE tasks, and the fifth question (RQ5) discusses the gaps and possible future improvements in this domain.

#### 2) SEARCH STRATEGIES

We used five digital libraries to conduct the automated search: Scopus, IEEE Xplore, ACM Digital Library, ScienceDirect, and SpringerLink. Scopus and ScienceDirect are general indexing systems that help to cover a broader scope for our search. On the other hand, IEEE Xplore, ACM Digital Library, and SpringerLink publish papers related to the most well-known conferences and journals related to the software domain.

The search string is built based on the following three key terms: "*Requirements Engineering*", "*Syntactic Processing*", and "*Semantic Processing*". These terms are derived from our main research question. Each of these terms is enriched by adding synonyms and sub-fields. Table 1 shows the whole set of selected keywords for this study divided into three groups: A, B, and C. These groups were used to create

**TABLE 1.** Keywords used in our study.

| Group | Main Keyword | Enriched List |
|---|---|---|
| Group A | Requirements Engineering | "requirement engineering" AND "software requirement" |
| Group B | Syntactic Processing | "syntax" OR "POS-tagging" OR "tagging" OR "Dependency Parsing" OR "shallow parsing" OR "chunking" OR "named entity recognition" |
| Group C | Semantic Processing | "semantic" OR "BERT" OR "word embedding" OR "word2vec" OR "Vector Space Model" OR "VSM" OR "Latent Semantic Analysis" OR "LSA" |

the final search query as follows:

$$A \textbf{ AND } (B \textbf{ OR } C)$$

#### 3) INCLUSION AND EXCLUSION CRITERIA

Inclusion and exclusion criteria are used to filter out papers that are not relevant to our research questions. We defined three inclusion criteria and four exclusion criteria.

##### *a: INCLUSION CRITERIA*

IC1: Peer-reviewed research presents an approach related to the field of software requirements engineering.
IC2: The research uses NLP techniques including syntactic or semantic processing.
IC3: It is published between 2010 and 2021

##### *b: EXCLUSION CRITERIA*

EC1: Secondary research is excluded (such as literature reviews, summaries, etc.)
EC2: The research is published in languages other than English
EC3: Duplicate papers (only the most recent and detailed one is considered)
EC4: The study does not provide detailed information about the proposed approach (such as, short papers, posters, etc.)

### *B. CONDUCTING THE SEARCH*

The review process consists of five main stages. Figure 1 illustrates these stages and the numbers of selected publications after each stage. Starting from the defined data sources, we obtained a total of 2,227 candidate papers. Duplicated papers were automatically eliminated using Parsifal tool[1] first, and then additional duplicate entries were manually eliminated by comparing authors, titles, and abstracts. After removing all duplicates, 1,573 papers remained. These papers are filtered by applying the selection criteria based on titles and abstracts. This stage led to the selection of 252 papers. A full-text review of those papers was then conducted to discard papers that did not satisfy our selection criteria. The remaining primary research papers after this stage are 98. Finally, reference checking led to an additional 6 relevant papers. After these stages, our final list consists of 104 unique primary papers.

[1] https://parsif.al

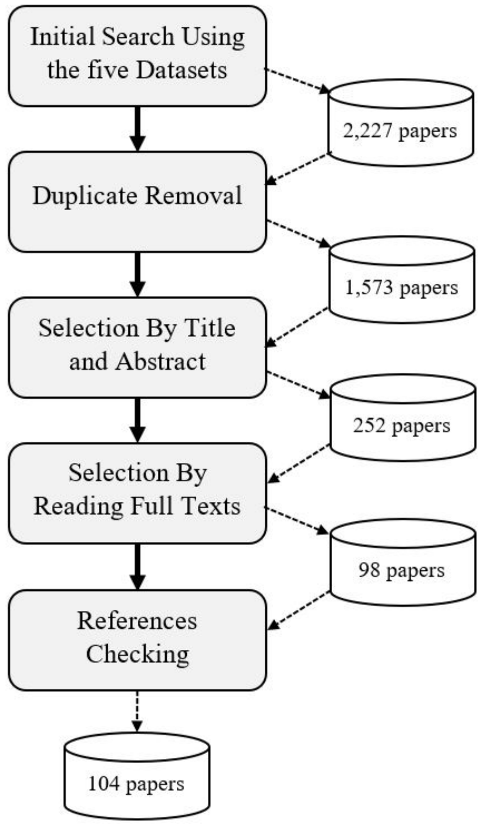



**FIGURE 1.** Search flow diagram for our systematic review.

### *C. DATA EXTRACTION AND CLASSIFICATION*

A Data Extraction Form (DEF) is developed to collect the required data to answer our RQs. The form is designed in a table format consisting of the following types of information:

- Bibliometric Information: author(s), publication year, type of publication, and publication venue.
- Targeted RE task(s).
- Proposed solution including the Syntactic and semantic information used to represent requirements.
- Evaluation Details: evaluation dataset, used metrics, and the results.
- limitations and constraints.

## V. RESULTS AND DISCUSSIONS

This section describes and explains the results obtained by the analysis of the 104 selected papers answering the RQs exposed in the previous sections.

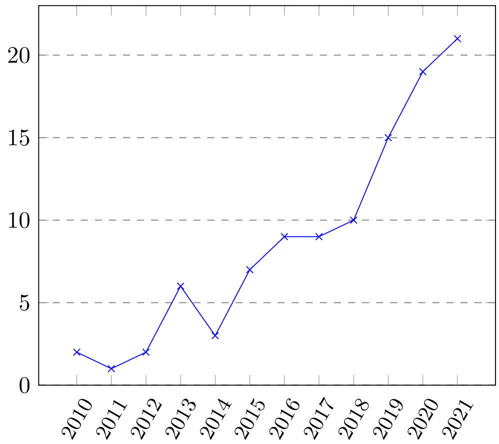



**FIGURE 2.** Number of published per year.

### A. (RQ1) WHAT IS THE STATE OF THE PUBLISHED LITERATURE ON RE WORKS THAT USE SYNTACTIC AND SEMANTIC REPRESENTATIONS FOR REQUIREMENTS?

As previously mentioned, our final list consists of 104 unique peer-reviewed research papers. Fig 2 shows the distribution of these publications per year. This chart reflects a growing interest in the use of syntactic and semantic levels of processing to handle RE tasks. The vast majority of papers (more than 85%) have been published since 2015.

Among the selected papers, 47% of them appeared in journals; 42% of papers were published in conference proceedings, while 11% of papers came from workshops.

The most popular venue for publishing articles related to our study is *IEEE International Conference on Requirements Engineering* and its workshops with more than 12% of papers, while the most popular journals are: *Information and Software Technology*, *Empirical Software Engineering*, *Requirements Engineering* with 5-6% of papers in each of them.

### B. (RQ2) IN WHICH RE TASKS ARE THE SYNTACTIC AND SEMANTIC ASPECTS MOSTLY CONSIDERED TO REPRESENT REQUIREMENTS?

We classified the retrieved papers based on the type of the targeted RE task. Fig. 3 shows the hierarchy of all 104 papers categorized into 6 top-level categories and 19 subcategories. The six main categories are:

#### 1) REQUIREMENTS ANALYSIS

This category represents the majority of publications (50 out of 104 papers). It includes papers focusing on the following RE tasks:

- **Requirements Classification:** Requirements classification is an important step towards automatically analyzing natural language requirements, especially when handling projects with large numbers of requirements [10]. We recognized 21 papers proposing solutions to various requirements classification tasks. Most of them focused on Functional/Non-Functional classification tasks [10], [11], [39]–[49], while the remaining focused on other classification tasks: security/Not security [50]–[52], topic-based classification [53], and classification based on requirements importance level [54].
- **Requirements Traceability:** Requirements traceability is used to capture the relationships between the requirements, the design, and the implementation of a system [55]. We recognized 21 papers related to this task, part of which focused on detecting the relationships between requirements (inter-dependency between requirements) [56]–[66], while the remaining focused on detecting the relationships between requirements and other artifacts (design documents and source code) [55], [67]–[74].
- **Requirements Clustering:** Requirements clustering is used to organize software requirements into a set of clusters with high cohesion and low coupling [75]. We recognized 5 papers presenting approaches to cluster requirements. These papers used the resultant clusters to understand the main functional groups or topics over requirements [76]–[78], to organize requirements in a tree structure (hierarchy) [79], or as a step towards discovering redundancy and inconsistency between requirements [80].
- **Semantic Role Labeling:** Semantic role labeling is the task of extracting semantic information from a software requirements specification [81]. Four papers addressed this task [81]–[83]. These works focused on mapping requirements to formal representations by extracting their main semantic elements such as actors, actions, and objects.

#### 2) REQUIREMENTS EXTRACTION

Requirements extraction (or elicitation) is one of the crucial steps in software development. We recognized 19 papers addressing tasks related to this category. These papers can be divided into three groups:

- **Extracting Requirements from Reviews and Forums:** 9 papers [63], [84]–[91] proposed solutions for this task, where various applications reviews and forums are used as a source for input texts such as App Store and Google Play.
- **Extracting Requirements from Textual Documents:** 7 papers [92]–[98] presented approaches to extract requirements from SRS documents, policies, user manuals, and emails.
- **Extracting Requirements from Similar applications:** 3 papers [99]–[101] focused on recommending requirements based on the specifications of similar applications. This goal is achieved by processing the description of similar products to suggest new possible features and generate creative requirements.

#### 3) QUALITY ASSESSMENT

Quality assessment tasks are concerned with detecting defects in software requirements specifications. [102], [103].


**FIGURE 3.** A chart illustrating the hierarchy of papers categorized based on their targeted RE task.

We recognized 17 papers focusing on tasks related to the following quality assessment tasks:

- **Ambiguity Detection:** This task helps in the identification of ambiguous requirements. The works under this sub-category (9 papers) can be further classified based on the discussed level of ambiguity:
  - Lexical Ambiguity: The main focus at this level is the ambiguity caused by words and terms. We recognized 6 papers related to this level [10], [93], [104]–[107].
  - Syntactic Ambiguity: This level focuses on detecting sentences that have different possible grammatical structures. We recognized one paper handling this level of ambiguity (Osama *et al.* [108]).
  - Semantic Ambiguity: This level focuses on detecting confusing contexts in sentences such as anaphoric ambiguity and coordination ambiguity. We recognized one paper handling this level (Ezzini *et al.* [109]).
  - Pragmatic Ambiguity: This level focuses on detecting sentences with multiple meanings. We recognized one paper handling this level (Ferrari *et al.* [110])
- **Incompleteness Detection:** This task is concerned with detecting any possible incompleteness in requirements statements. We recognized 3 papers handling this task [111]–[113].
- **Conformance With Templates:** This task focuses on verifying that the requirements are indeed written according to pre-defined templates. We recognized one work proposing a solution to handle this task (Arrora *et al.* [114]); specifically, this work focused on checking the conformance of requirements with two well-known templates: Rupp [115] and EARS [116] templates.

- **Vagueness Detection:** Vagueness occurs when a statement can have a continuum of interpretations (e.g., when using words like tall, large, etc.). We recognized one paper focusing on this problem (Cruz *et al.* [117]).
- **General Assessment:** Other papers suggest approaches providing a general assessment for different aspects of requirements quality. Three papers can be classified under this type [102], [103], [118]

#### 4) MODELING

Modeling software requirements is the process of transferring the natural language requirements into models and diagrams [119]. We recognized 12 papers proposing solutions related to this type of RE tasks. One of the main differences between these works is the type of the generated model:

- Use Case Diagrams [119]–[122], that focus on actors and their corresponding actions.
- Feature Models [123], [124], that define features and their dependencies.
- Conceptual Diagrams [125], [126], that focus on concepts and relationships between them.
- Glossaries [127], [128], which define technical terms which are specific to an application domain.
- Goal Models [129], that focus on the objectives which a system should achieve through the cooperation of actors in the intended software and the environment.
- Semantic of Business Vocabulary and Rules (SBVR) [130], which defines the semantics of business vocabulary, business facts, and business rules.

#### 5) REQUIREMENTS MANAGEMENT

Requirements management is an ongoing activity throughout the development process [131]. We classified two tasks under this category:

- **Requirements Prioritization:** the main target for this task is to determine which candidate requirements of a software product should be included in a certain release. We recognized 5 papers handling this task [132]–[136].
- **Effort Estimation:** This task focuses on estimating the effort involved in implementing a requirement [137]. We found 2 papers proposing methods to handle this problem [137], [138]. Both of them focused on estimating the effort in Agile development methodologies.

#### 6) OTHERS

One of the retrieved papers (Mishra *et al.* [139]) focused on building a language model for the software requirement domain. This model was based on a domain-specific text corpus collected by crawling the software engineering category on Wikipedia.

### C. (RQ3) WHAT ARE THE PROPOSED REPRESENTATIONS IN THE LITERATURE FOR RE TASKS?

In general, almost all covered papers consist of two main phases:

(1) Representation Phase: NLP processing steps are applied to analyze requirements texts, and to capture linguistic information in order to represent them in various forms.

(2) Solving Phase: the results of the previous stage are used to solve the targeted problem based on various ML and non-ML approaches.

Fig. 4 shows these two phases and the techniques used for each of them. Overall, we recognized **five** different text representation techniques used to handle the first phase: Lexical and syntactic features, ontology-based representation, VSM, topic modeling-based representation, and advanced embedding techniques. On the other hand, two families of solutions were proposed for the second phase: Machine Learning approaches which represent 55% of papers, and Rule-Based approaches (in 45% of papers) where patterns, regular expressions, and heuristics were used.

To answer our third research question (RQ3), we explore the representations used in the first phase in more detail:

#### 1) LEXICAL AND SYNTACTIC FEATURES

This part represents the largest group with more than 46% of papers (48 out of the selected 104 papers). The solutions proposed in these papers share a similar pipeline: (1) Applying a set of NLP pre-processing steps. (2) Representing each requirement as a pre-defined set of linguistic features. (3) Then, proceeding to the solving phase which uses ML techniques (such as Decision Trees [52], [81], [97], [118], [138], SVM [10], [53], [88], and RF [49], [60], [90], [96]), or rule-based approach using a set of syntactic regular expressions [95], [117], [125], [126], [130].

The used pre-defined set of features (in step 2) usually includes features related to the following four groups:

- **The lexical and morphological features**, which are based on requirement words and their morphological information, such as token, stem (or lemma), n-gram (sequence of words).
- **The syntactic features**, which are derived from the syntax-related information in the requirement statement, such as POS-tag, chunks (noun or verb phrases), dependency relations, and the entities extracted using NER.
- **The frequency-based features**, which are based on the requirements list frequency metadata such as the number of words and the number of requirements.
- **Dictionary-based features**, which are extracted with the help of special dictionaries representing special words lists.

To further investigate the commonly used features, Fig. 5 shows the most frequently used features and the number of papers that consider each of these features. The most used feature is the POS-tag, as it is considered in 34 out of the total 48 papers related to this type of representation.

#### 2) ONTOLOGY-BASED REPRESENTATION

This group represents about 9% of papers (9 out of 104). It benefits from pre-defined lexicons and semantic resources

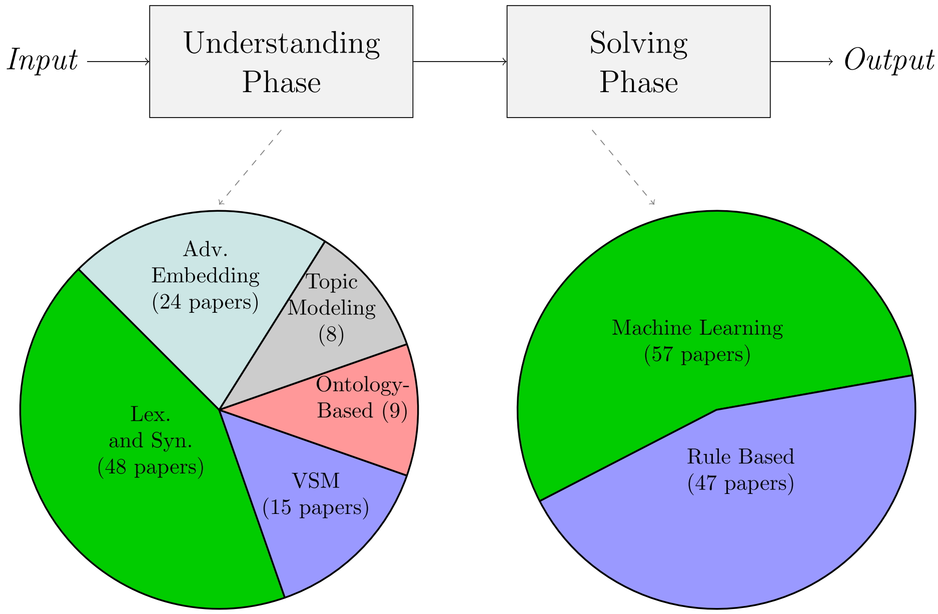



**FIGURE 4.** The general flow and the used techniques for each phase.

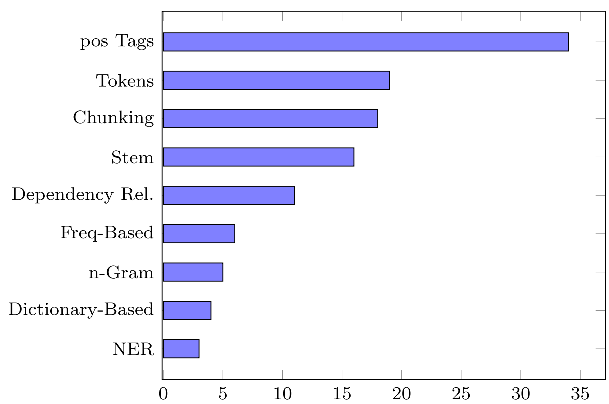



**FIGURE 5.** Frequently used features.

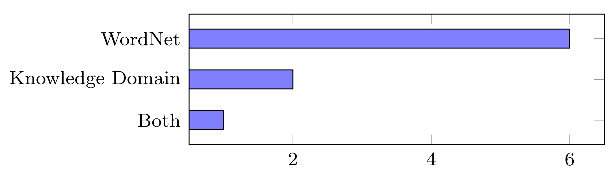



**FIGURE 6.** The used ontologies with their frequencies.

to extend the lexical and syntactic features. Most of these papers used a rule-based approach in their solutions by applying various ontology-path-based similarity measures. [74], [105], [109], [110], [113].

One of the main differences between these papers is whether the used ontology is general or domain-specific. Fig. 6 shows three types of works under this category based on the type of the used ontology: 6 papers used a general ontology (specifically WordNet) [74], [105], [113], [127], [129], [140], while 2 papers used a domain-specific ontology [109],

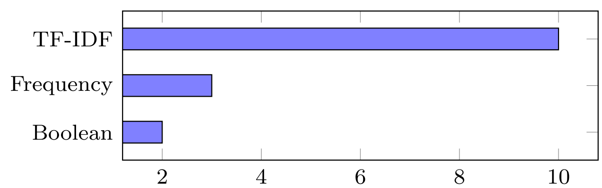



**FIGURE 7.** Weighting methods used in VSM representations with their frequency.

[110], and one paper merges both approaches in its similarity calculations [83].

### 3) VSM

Various forms of VSM representation are used in 15% of papers (15 papers). These works combine the BOW technique with different weighting methods. The most frequent combination is BOW with TF-IDF (10 papers). Fig 7 shows all used weighting methods with their frequencies.

Based on this type of representation, each requirement is represented as a vector of words or tokens or stems combined with their weights. These vectors are used in the subsequent phase to apply similarity-based rules [55], [71] or as an input to train machine learning models (such as SVM [40], [69], [87], [141]).

### 4) TOPIC MODELING BASED REPRESENTATIONS

About 8% of papers (8 papers) used topic modeling techniques to represent requirements texts based on automatically discovered latent topics. Two main topic modeling techniques are used in these papers: LSA and LDA. Fig 8 shows the distribution of these two techniques over the related papers.

The output of both techniques is a k-dimensional space that reflects a k latent topics within the processed requirements.

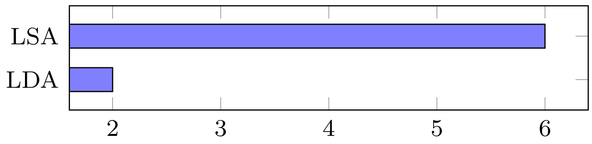



**FIGURE 8.** The used Topic modeling techniques with their frequency.

**TABLE 2.** The used word embedding techniques with their related papers.

| Word Embedding | Related Papers |
|---|---|
| word2vec | [43], [50], [58], [59], [67], [76], [101], [104], [106], [139] |
| BERT | [11], [47], [56], [64], [84], [91], [98] |
| Embedding Layer | [39], [51], [112], [137] |
| GloVe | [54] |
| FastText | [66], [128] |

Then, each requirement is represented as a vector representation in this k-dimensional space. In most related papers, this representation is employed to calculate the similarity between requirements as a part of similarity-based rules [70], [107] or clustering machine learning techniques [78], [99], [135].

### 5) ADVANCED EMBEDDING REPRESENTATIONS

This part represents the second largest group of papers (24 papers) with a major increase in the last couple of years. It is noted that papers that use this kind of representations achieve promising results in many major tasks such as requirement classification [11], [50], [51], traceability [56], [58], [59], [64], ambiguity detection [106], and requirement extraction [91], [98].

Based on this type of representation, each requirement is represented as a vector of floating point values. The proposed representations in the related papers are built following two generic steps (Fig. 9): First, each token in the requirement text is represented using a word embedding technique; then the final requirement representation is constructed using the representation of its words. To further investigate papers that use advanced embedding techniques, we explore the techniques and models used in the implementation of each of these two steps.

- **Word Representation:** Various word embedding techniques were used in the related literature. Word2vec and BERT are the most common techniques utilized in this group; together they were used in 17 out of the total 24 papers related to this group. Other works used GloVe and FastText models, while the remaining papers used an embedding layer instead (added to the front of their neural network model).

  Table 2 shows the related papers for each of the used word embedding techniques.
- **Statement Representation:** Papers that use this type of representation can be further classified based on the method used to merge word embeddings in order to represent statements. Table 3 shows all related papers classified into three types of statement embedding techniques.

**TABLE 3.** the used statement embedding techniques with their related paper.

| Statement embedding | Related Papers |
|---|---|
| Aggregation-based | [11], [47], [56], [58], [59], [64], [66], [76], [84], [91], [98], [101], [104], [128] |
| RNN-based | [43], [51], [54], [67], [112], [137] |
| CNN-based | [39], [50] |

*The aggregation approach* is the most used technique (16 out of 24). It produces the requirement representation by applying various aggregation methods on word embedding results such as:

– Using the average of words embedding to represent statements [66], [101], [128].
– For BERT-based word embedding, a special token ([CLS]) is added to the beginning of the sentence and is used to calculate an aggregated representation of the statement [11], [64], [84], [91], [98].
– Combining the syntactic information with the semantic ones by (1) classifying words based on their syntactic role, (2) calculating the weighted average for each role, then (3) concatenating the resultant vectors to form the final representation [58].
– Representing requirements by representing its main "semantic frames" which can be retrieved with the help of FrameNet [142]. The representation of these frames is calculated by averaging the embedding vectors of their words [59].

*Recurrent Neural Network (RNN) based representation* takes word embedding vectors as input. It finds a dense and low-dimensional semantic representation for each requirement statement by sequentially and recurrently processing its words. Many RNN architectures have been used in the related papers such as LSTM [43], [51], [137], Bi-LSTM [54], BI-GRU [67], and Skip-Thought [112].

*Convolutional Neural Network (CNN) based representation* takes word embedding vectors as input. It finds the final representation through a number of convolutional layers, pooling layers, and fully connected layers [39], [50].

## D. (RQ4) WHAT ARE THE MAIN RESEARCH DIRECTIONS TO REPRESENT REQUIREMENTS FOR EACH CATEGORY OF RE TASKS?

To answer this question, we start with a general overview of how the research directions to represent requirements have evolved over the last decade; then we present a deeper analysis of the trends and possible future directions in each category of RE tasks.

### 1) GENERAL OVERVIEW

Fig. 10 shows the number of papers according to the used NLP representation and the year of publication. The bubble

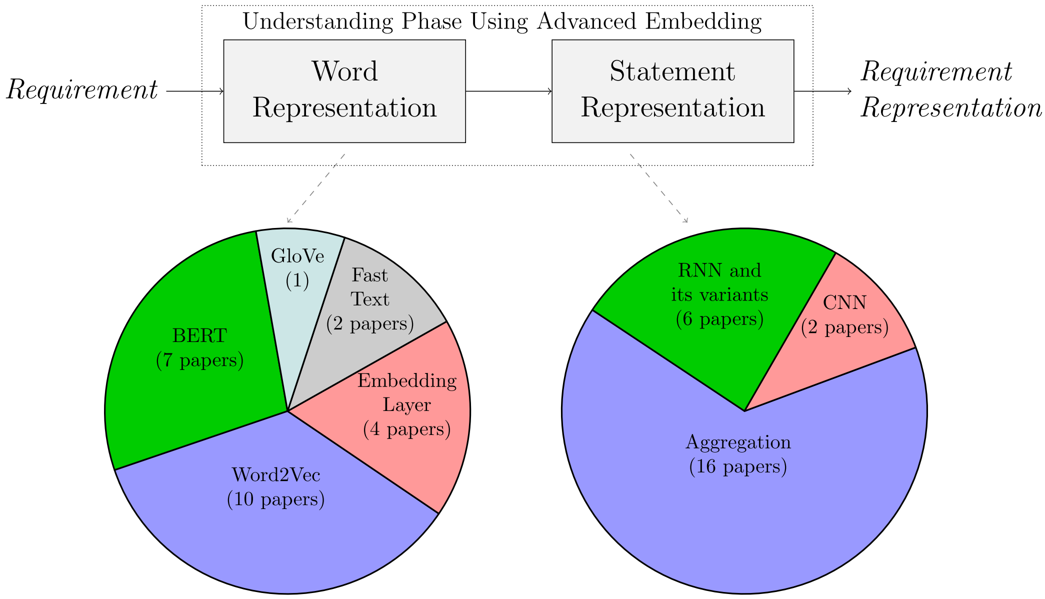



**FIGURE 9.** The general flow of representing requirements based on advanced embedding techniques.

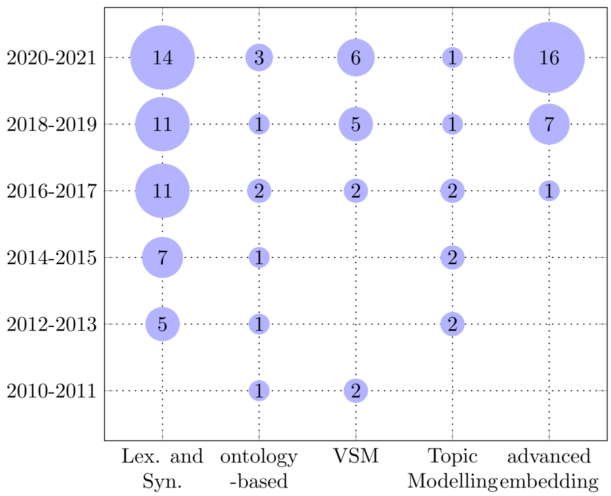



**FIGURE 10.** Bubble chart showing the number of papers according to the used NLP representation and the year of publications.

chart (Fig. 10) shows clearly how the trend has changed in the last few years from the lexical and syntactic features to the advanced embedding techniques. These embedding representations became the most used type in the last 2 years (2020-2021) with 15 out of 40 published works in that period.

### 2) RESEARCH DIRECTIONS FOR EACH CATEGORY OF RE TASKS

To further investigate the trends of various representations, we studied their distribution over the main categories.

Fig. 11 shows the number of papers that used each of these representations for each category. As noted, representing requirements based on a set of lexical and syntactic features is the most used approach in many tasks, especially

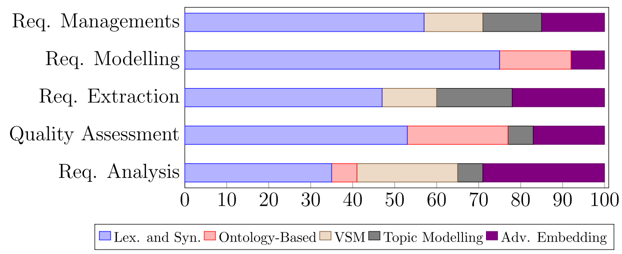



**FIGURE 11.** The percentage of used NLP representation for each category of RE tasks.

in quality assessment and requirement modeling. However, there is a clear interest in the use of advanced embedding and VSM-based representations in requirement analysis tasks.

The following sub-sections discuss the trends for each category.

- **Requirement Analysis:** Embedding techniques have been proven to be useful for various requirement analysis tasks [11], [58], [59], [64]. Two recent studies [11], [64] compared the performance of lexical-syntactic features representation with advanced embedding representation, and concluded that using embedding vectors led to better results in requirement classification [11] and traceability detection [64] tasks. This finding is consistent with the results observed in the NLP literature [143], which indicate that embedding techniques are powerful in determining the similarity of words and sentences. of words and sentences. Similarity represents the core operation in almost all requirements analysis tasks.
- **Quality Assessment:** More than 50% of quality assessment papers used lexical and syntactic features to represent requirements (Fig. 11). However, a deeper analysis

of related papers showed that we can differentiate between two types of quality-related RE tasks:

- The majority of papers focused on lexical and syntactic quality-related issues such as lexical ambiguity and conformance with templates. Many significant contributions provide experiments that indicate the effectiveness of using lexical and syntactic features to represent requirements when handling this category of tasks [102], [103], [114], [144].
- The remaining papers focused on semantic quality-related issues such as incompleteness, and semantic-level quality problems. For these tasks, the most used representation in the literature is ontology-based representation [109], [111], [113]. Besides, one of the recent interesting works employed advanced embedding representation based on an embedding layer with a Skip-Thought model to handle incompleteness detection problem [112].

- **Requirements Extraction:** It is noted that there are two directions to handle requirements extraction tasks based on the source of data:
  - Extracting requirements from documents: the structured format of these documents motivated many researchers to use lexical and syntactic features. All retrieved papers under this sub-category used lexical and syntactic representations [92], [95], [96].
  - Extracting requirements from reviews, forums, or similar applications: Most of the recent papers (especially the ones published in the last 2 years) used advanced embedding techniques to handle these tasks [84], [91], [101]. This finding could be due to the approaches used to handle this type of requirement extraction tasks; most solutions handle the problem as a classification task (classify sentences or phrases into requirements or not requirements classes) or a clustering task (group sentences that mention the same aspects in the product). Both approaches (clustering and classification) make the embedding technique more suitable (as mentioned previously when discussing the requirement analysis case).
- **Requirements Modeling:** Almost all works under this category represent requirements based on their lexical and syntactic features or using ontology-based representation [120], [129], [130]. These features are used in the form of rules and syntactic patterns to generate various models based on rule-based transfer techniques.
- **Requirements Management:** Most of the papers under this category use lexical and syntactic features representation [133], [134], [136]. However, the number of NLP-based publications related to this category is still limited to have a clear conclusion regarding the suitable representations for this type of tasks.

### *E. (RQ5) WHAT GAPS AND POTENTIAL FUTURE DIRECTIONS EXIST IN THIS FIELD?*

Our survey shows that the use of NLP-based requirements representation has made remarkable progress over the last decade. Based on our findings in the previous research questions (especially RQ4), the use of advanced embedding techniques to represent requirements seems to be promising for most RE tasks (mainly requirement analysis, extracting requirements from reviews and forums, and semantic-level quality tasks). Besides, representing requirements based on a set of lexical and syntactic features or using ontology-based information seems more suitable for other RE tasks (such as modeling and syntax-level quality tasks).

In the following paragraphs, we will discuss the gaps that represent potential future research directions in this domain.

#### 1) MORE RE DOMAIN-SPECIFIC WORD EMBEDDING MODELS

As shown in Fig. 9, word representation plays an essential role in representing requirements (mainly when using embedding techniques). To implement this step, most researchers use generic word embedding pre-trained models in their solutions. One of the main limitations of these generic models is their inability to provide appropriate representations for many domain-specific words. For example, the domain-specific meaning of words like "virus," "cookies," "Python," "fork," etc. can not be captured based on models trained on generic corpora [139]. Building such domain-specific word embedding models is notoriously a challenging task, especially with the little size of data available in the RE domain. One of the recent works [139] provided an embedding model trained on 92 MB of texts collected from Wikipedia pages related to the software engineering domain. However, more research is needed (1) to have embedding models trained on more practical industrial texts, (2) and to evaluate the use of these models in various RE tasks.

#### 2) MORE SUITABLE STATEMENTS REPRESENTATION TECHNIQUES

Another important challenge, related to embedding techniques, is the way of merging word vectors to formulate requirements representations (step 2 in Fig 9). Although the suggested techniques lead to good accuracy in some RE tasks (such as requirements classification and traceability detection), these representations can still be considered as a step towards more suitable representations. Most of these works use aggregation techniques (that ignore word order [145]) or RNN-based technique (that usually focus on predicting the next element in a sequence [146]). These works can be further improved by considering more RE domain-specific representations [58]. One of the main potential future directions to handle this problem is using a "syntax-aware sentence embedding" technique that considers both the main

**TABLE 4. The final list of selected papers categorized based on the type of targeted RE task and the used text representation.**

| | Lexical and Syntactic Features | Ontology-Based | VSM | Topic Modeling-Based | Advanced Embedding |
|---|---|---|---|---|---|
| Requirement Analysis | [152], [44], [65], [97], [53], [46], [10], [81], [80], [60], [57], [52], [61], [45], [77], [82], [72] | [74], [140], [83] | [41], [55], [48], [40], [51], [79], [71], [73], [141], [69], [42], [49], [62] | [68], [78], [70] | [39], [50], [98], [43], [76], [64], [58], [11], [67], [66], [47], [54], [56], [59] |
| Requirements Extraction | [89], [96], [94], [153], [90], [95], [88], [92], [97] | | [87], [100] | [99], [85], [86] | [63], [91], [101], [84], [98] |
| Quality Assessment | [103], [117], [114], [118], [144], [93], [108], [111], [102] | [113], [105], [110], [109] | | [107] | [112], [106], [104] |
| Modeling | [123], [130], [125], [120], [121], [124], [122], [119], [126] | [129], [127] | | | [128] |
| Management | [138], [134], [136], [133] | | [132] | [135] | [137] |
| Others | | | | | [139] |

syntactic roles in requirement statements (such as actor, action, objects,...) and the semantic aspects of requirements.

### 3) MORE ADAPTIVE SYNTACTIC PROCESSING

Many studies have concluded that there is a gap between RE tasks automation research and its implementation undertaken in industrial and real-life projects [147]–[149]. This gap is more obvious in the case of rule-based approaches since they usually need the requirements to be represented based on specific templates [150]; hence, their success strongly depends on the consistency of the requirements with the predefined templates [114], [150]. Using such "template-based" approaches to handle RE tasks could lead to lower accuracy when applied to new requirements written based on some variations of the predefined template or a completely different template. These situations are common in real-life projects when it is hard to control requirements authoring environments, especially in large development projects, or when one has little control over the requirements authoring environments. [12], [114], [147]–[151]. One of the potential future research directions is working on more adaptive approaches which can be more flexible when handling requirements, such as identifying the syntactic structures automatically and building more adaptive approaches based on the dynamically identified structure.

### 4) MORE RESEARCH IN SOME RE TASKS

Only a few works handled the following RE tasks: requirements prioritization, effort estimation, and semantic and pragmatic level quality tasks. More research should be conducted to explore the efficiency of using advanced embedding techniques for these tasks.

## VI. MAIN FINDINGS

In this section, we summarize the main conclusions and findings in the following points:

(i) There is a significant increase in the number of papers that use NLP-based requirements representation over the last decade. The increase is more obvious in the last 2 years.

(ii) We recognized 104 papers that employed an NLP-based representation to solve various RE tasks. We classified these works based on (1) the targeted RE task using a hierarchy consisting of 6 top-level categories and 19 sub-categories (Fig. 3), (2) the used NLP-based representation using 5 classes of text representation techniques (Fig. 4).

(iii) Using advanced embedding techniques to represent requirements seems to be promising for most RE tasks (especially requirement analysis, extracting requirements from reviews and forums, semantic-level quality tasks). Besides, representing requirements based on a set of lexical and syntactic features or using ontology-based information seems more suitable for other RE tasks (such as modeling and syntax-level quality tasks).

(iv) Most of the proposed embedding-based representations use generic language models that suffer from limitations when representing the meaning of software engineering terms. Having more domain-specific language models to represent words is considered one of the important research directions in future works.

(v) Most of the proposed embedding-based techniques represent requirements statements without giving attention to the special structure of a software requirement that consists of clear components (actor, action, data object, etc). Using syntax-aware statements embedding could be one of the possible solutions to have more domain-specific representations that reflect the semantics of requirements in a better way (compared to traditional aggregation approaches).

**TABLE 5. The Detailed Info for all 104 selected papers.**

| Paper | Targeted Task | Input | Output | Representation Method | Solving Method | Text Processing | Solving Method Details | Dataset | Results |
|---|---|---|---|---|---|---|---|---|---|
| Baker et al. 2019 [39] | Classification | a Req. | NFR classes | Advanced Embedding | ML | Stop words removal; Tokenization; Stemming; | ANN; CNN* | PROMISE+ | F1: 0.82-0.92 |
| Dias et al. 2020 [40] | Classification | a Req. | FR/NFR; NFR classes | VSM | ML | Stop words removal; Tokenization; Stemming; BOW, TFIDE, CH2 feature selection | SVM*; MNB; KNN; LR | PROMISE (lima version) | F1: FR/NFR 0.91 / NFR: 0.72 |
| Casamayor et al. 2010 [41] | Classification | a Req. | NFR classes | VSM | ML | Sop words removal; Stemming; BOW + TF-IDF | Bayesian Classifier +EM strategy | PROMISE | Acc: 0.75 |
| Dalpiaz et al. 2019 [10] | Classification | a Req. | FR/QR | Lexical-Syntactic features | ML | 17 lexical-syntactic features including syn. dependency-based features | SVM | PROMISE+ | F1: F(0.79); Q(0.76) |
| Hey at al. 2020 [11] | Classification | a Req. | FR/QR; NFR classes; FR classes | Advanced Embedding | ML | BERT | FFNN | PROMISE (Dalpiaz Version) | FR/QR up to 0.94; NFR 0.87; FR up to 0.92 |
| Khatian et al. 2021 [42] | Classification | a Req. | NFR classes | VSM | ML | Stop words removal; Stemming; BOW | DT; KNN; RF; NB; LR* | PROMISE | F1: 0.75 |
| Rahman et al. 2019 [43] | Classification | a Req. | NFR classes | Advanced Embedding | ML | word2vec | LSTM; GRU; CNN | PROMISE | F1: 0.71 |
| Rashwan et al. 2013 [44] | Classification | a Req. | NFR classes | Lexical and Syntactic Features | ML | Tokenization, Sentence splitter, Stemming; | SVM | CONCORDIA +PROMISE | F1: 0.67-0.84 |
| Younas et al. 2020 [45] | Classification | a Req. | NFR classes | Lexical and Syntactic Features | Rule-Based | stopwords removal, stemming, POS-tagging, famous NFR indicator keywords | similarity-based approach | PROMISE | F1: 0.64 |
| Mahmoud et al. 2016 [46] | Classification | a Req. | NFR classes | lexical and syntactic features | Rule-Based | lemmatization; Normalized Google Distance (NGD) | similarity-based approach | Own Dataset | R: 0.88 /P: 0.52 |
| Kici et al. 2021 [47] | Classification | a Req. | FR/NFR | Advanced Embedding | ML | word embedding (DistillBERT and BiLSTM + ELMo) | multi-class text classification | DOORS + PROMISE | F1: 0.80 |
| Raharja et al. 2019 [48] | Classification | a Req. | Quality aspects | VSM | ML | Tokenization, Stop words removal, Stemming, TF-IDF | Fuzzy similarity measure + KNN | 6 datasets (PROMISE+) | P:0.68; R:0.55 |
| Chatterjee et al. 2020 [54] | Classification | a Req. | Significant FR. | Advanced Embedding | ML | GloVe trained on 124 SRS | Bi-LSTM + Attention | Own Dataset | F1: 0.86 |
| Palacio et al. 2019 [50] | Classification | a Req. | Sec/NSec | Advanced Embedding | ML | word2vec trained on a collected security dataset | CNN | Own Dataset | Acc: 0.71-0.96 |
| Kobilica et al. 2020 [51] | Classification | a Req. | Sec/NSec | VSM | ML | BOW | 22 ML algorithm (including LSTM*) | SecReq | Acc: 0.84 |
| Li et al. 2020 [52] | Classification | a Req. | Sec/NSec | Lexical and Syntactic Features | ML | 140 Keywords (Lexical features) + 34 linguistic rules (Syntactic Features) | J48; NB; LR | SecReq +PROMISE | F1: 0.63 |
| Silva et al. 2020 [141] | Classification | a Req. | Design Patterns | VSM | ML | TF-IDF | LR, MNB, SVM*, RF | PROMISE+ | F1: 0.52 |
| Ott at al. 2013 [53] | Classification | a Req. | Req. Topic | Lexical and Syntactic Features | ML | Tokenization, n-gram, Surrounding requirements | MNB, SVM* | DCU | R:0.8/ P: 0.6 |
| Krasniqi et al. 2021 [49] | Classification | Issues | quality related issue | VSM | ML | Length-based features, BOW, n-gram, syntactic features, Semantic Triplet | RF | Own Dataset | F1:0.79 |
| Gulle et al. 2020 [76] | Clustering | a set of Req. | Clusters | advanced embedding | ML | Tokenization; Stop words removal; Remove Template Words; word2vec | Word Mover's Distance based clustering | CrowdRE | no measure exists |
| Casamayor et al. 2012 [77] | Clustering | use-cases | Clusters for functional group | lexical and syntactic features | ML | POS-tagging; chunking; categorize candidate responsibilities based on the NP and V relates to. | Clustering Based on EM, CobWeb, Xmeans, DBScan | Own Dataset | Rand Index: 0.80 |
| Misra et al. 2016 [78] | Clustering | SRS Document | Clusters | Topic modeling | ML | pos tagging; chunking; Remove stopwords; lemmetization; LSA | Theme based clustering | Own Dataset | Purity: 0.46; F1: 0.52. |
| Eyal et al. 2018 [79] | Clustering | a set of Req. | Clusters | VSM | ML | Tokenization; stopwords removal; stemming; VSM | Agglomerative Hierarchical clustering (AHC) | Own Dataset | P:0.72-0.83/ R:0.54-0.61 |
| Mezghani et al. 2018 [80] | Clustering | SRS Document | Clusters | lexical and syntactic features | ML | pos tagging; NP chunking to detect technical business terms | K-mean | Own Dataset | No Available Results |
| Das et al. 2021 [56] | Traceability | Two Reqs | Similarity score | Advanced Embedding | Rule-Based | BERT | Cos Similarity | 800 pairs of reqs | Acc: 0.88 |
| Blake et al. 2015 [57] | Traceability | Two SRS documents | identify overlapping requirements | lexical and syntactic features | Rule-Based | Tokenization; stopwords removal; generate synonyms; determine verbs | Similarity between reqs is calculated based on number of similar words | Own Dataset | Not Available |
| Sonbol et al. 2020 [58] | Traceability | Two Reqs | similarity score | Advanced Embedding | Rule-Based | Tokenization; POS Tagging; NP chunking; classify requirements to semantic dimensions; word2vec | Manhattan Distance Based Similarity | 5,852 pairs of reqs | F1: 0.92 |
| Alhoshan et al. 2019 [59] | Traceability | Two Reqs | Similarity score | Advanced Embedding | Rule-Based | tokenisation, stop-word removal, POS tagging and lemmatisation, Embedding-based representation of semantic frames | relatedness score based on frame Embeddings and cos sim | 1,770 pairs of reqs | F1: 0.86 |
| Deshpande et al. 2019 [60] | Traceability | Two Reqs | Dependency and its type | lexical and syntactic features | ML | tokenized, stop words removal; lemmatization | RF, SVM, NB | PURE dataset | F1:0.89 |
| Samer et al. 2019 [61] | Traceability | Two Reqs | related / not related | lexical and syntactic features | ML | Stop words Removal , merged synonyms, lemmatization | Linear SVM, NB,RF*, and KNN | OpenReq Dataset | F1: 0.89 |
| Sultanov et al. 2011 [62] | Traceability | Two Reqs | related / not related | VSM | ML | stop words removal, stemming, VSM+TF-IDF; agent words | swarm intelligence | Own Dataset | F1: 0.58 |
| Lu et al. 2017 [63] | Traceability | Two Reqs | related / not related | lexical and syntactic features | rule based | (A) Jaccard sim based approach (B) stop words removal; stemming; SVM+Freq; Cos similarity | apply two approaches and use fuzzy logic to merge results | Own Data from 14 projects | Acc: 0.83 |
| Deshpande et al. 2021 [64] | Traceability | two Reqs | related / not related | advanced embedding | ML | stop words removal; tokenization; lemmatization; BERT | BertForSequence Classification | Redmine Dataset | F1: 0.93 |
| Guo et al. 2017 [67] | Traceability | SRSs and Design documents | related / not related | advanced embedding | ML | word2vec trained on a domain corpus | RNN (Bidirectional Recurrent Gated Unit) | Own Dataset | Mean Average Precision: 0.59 |
| Chhabra et al. 2017 [73] | Traceability | use cases+code | code-reqs links | VSM | rule based | VSM based model | a set of huristics | Own Dataset | F1: 0.56 |

**TABLE 5.** ***(Continued.)*** **The Detailed Info for all 104 selected papers.**

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Wang et al. 2021 [55] | Traceability | reqs and artifcats | trace links | VSM | rule based | normalization, stemming, VSM, word2vec | ranking technique | Easyclinic, iTrust, eTour | Not Available |
| Wang et al. 2020 [68] | Traceability | reqs and artifcats | trace links | Topic modeling | ML | stop words removal, pos tagging, stemming, TF–IDF, Topic Modeling | Biterm Topic Model–Genetic Algo | Own Dataset | R/P larger than 0.70/0.30 |
| Rasekh et al. 2019 [69] | Traceability | reqs and artifcats | trace links | VSM | ML | TF-IDF+BoW | Bayesian learning , RBF Network, LR, SVM, DT | Easyclinic, albergate | P:0.91; R:0.94 |
| Mahmoud et al. 2015 [70] | Traceability | reqs and artifcats | trace links | Topic modeling | rule based | VSM, VSM with thesaurus support (VSM-T), POS-enabled VSM (VSM-POS), latent semantic indexing (LSI), LDA, explicit semantic analysis (ESA), and normalized Google distance (NGD). | similarity function | CM-1, eTour, and iTrust. | VSM-T, VSM-POS, ESA, and NGD outperform LSA and LDA. |
| Alia et al. 2019 [71] | Traceability | reqs and artifcats | trace links | VSM | rule based | pos tagging, stemming, | apply baseline approaches + Constraint-based pruning to approve links | iTrust, Pooka, Lynx, SIP | 11%-107% and 8%-64% higher P and R than VSM and JSM |
| Aldekhail et al. 2022 [65] | Traceability | a set of reqs | conflicts | lexical and syntactic features | rule based | basic NLP processing | rule based system to identify conflicts | CPMS, ENP, FMS | Accuracy: 100% |
| Shah et al. 2021 [140] | Traceability | a set of reqs | conflicts | ontology-based | rule based | tokenization, pos tagging, parser, dependency tree, WordNet | clustering requirements, then process clusters to detect conflicts | 5 datasets | F1:0.65 |
| Arora et al. 2015 [74] | Traceability | set of reqs+change senario | inter-requirement dependency | ontology-based | rule based | chunking; extract NP and VP; similarity measure using Levenstein and path based sim | similarity based algorithm | Two industrial case studies | detected 99% of the impacted reqs |
| Nicholson et al. 2021 [66] | Traceability | a set of reqs | dependencies | advanced embedding | ML | word embedding | LR*, RF, NN | 3 test cases | F1: 0.55-0.67 |
| Leitao et al. 2021 [72] | Traceability | Design Spec. and Reqs | Association Rules | lexical and syntactic features | ML | tokenization, pos tagging, chunking | Association Rule Mining | Own Dataset | 238 rules were extracted |
| Diamantopoulos et al. 2017 [82] | Semantic Role Labeling | a Req | Concepts + Relations | lexical and syntactic features | ML | Fetures: lemma, pos, syntactic dependency relations | logistic regression classifier | Own Data | F1: 0.76 |
| Wang et al. 2016 [83] | Semantic Role Labeling | a Req | semantic roles | ontology-based | ML | Fetures: lemmas; pos tags; chunking; Verb class; parsing tree; word sense fetures using wordNet and domain ontology | The maximum entropy classifier | Own Dataset | F1:0.85 |
| Wang et al. 2015 [81] | Semantic Role Labeling | SRS | semantic roles | lexical and syntactic features | ML | tokenization, pos tagging, parsing, NER | DT classifier | 18 SRS | P: 0.92-0.93; R: 0.91-0.92 |
| Ezzini et al. 2021 [109] | Ambiguity Detection | a Req | is ambigious? | ontology-based | rule based | Tokenizer, POS Tagger, Lemmatizer, parser, domain-specific corpus based on wiki and extracted nouns | a set of rules and heuristics | 20 Reqs documents | P: 0.80; R: 0.89 |
| Wang et al. 2013 [93] | Ambiguity Detection | a set of Reqs | Overloaded and Synonymous ambiguity | lexical and syntactic features | rule based | lexical features, contextual features, pattern based features | C-value method to extract candidate concepts | 4 projects (459 reqs) | Mean average precision : 0.52-0.57 |
| Osama et al. 2020 [108] | Ambiguity Detection | a Req | resolving syntactic ambiguity | lexical and syntactic features | rule based | syntactic parsing | set of heuristics | 126 requirements | P:0.65 precision, R:0.99 |
| Misra et al 2013 [107] | Ambiguity Detection | a Req | aliases disambiguity | Topic modeling | rule based | POS Tagging, terms extraction, stop words removal, misspelling identification, aliases identification, LSA | Similarity-based approach | 65 reqs | F1: 0.6 |
| Ferrari et al. 2012 [110] | Ambiguity Detection | a set of Reqs | detect ambiguity | ontology-based | rule based | Build a domain related knowledge graphs | takes the concepts, searches for concept paths within each graph, compare paths | proof of concept with an example | - |
| Mishra et al. 2019 [104] | Ambiguity Detection | 2 corpuses: related/not related to the domain | domain specific ambiguous CS words | advanced embedding | rule based | Tokenization, Punctuation Removal,Stop word removal, PoS Tagging, Lemmatization, learn word2vec from each corpus | find out which of the commonly used CS words have a different meaning | a set of examples | - |
| matsuoka et al. 2011 [105] | Ambiguity Detection | SRS | detect ambiguous terms | ontology-based | rule based | sentence tokenization, extract terms using wordNet, C-Value | For each term: get related sentences, and use C-value and WordNet semantic similarity to cluster them | a set of examples | - |
| Ferrari et al. 2019 [106] | Ambiguity Detection | corpuses from different domains | ambiguous terms | advanced embedding | rule based | crawl Wikipedia to extract domain specific documents; Language Models Generation | For each domain: find most freq. nouns, find a set of top similar words for each noun. Compare similar words to find ambiguity score | data from 5 domains | Maximum Kendall's Tau of 88 |
| Dalpiaz et al. 2019 [144] | Ambiguity Detection | a set of Reqs | near-synonyms terms | lexical and syntactic features | rule based | lexical and syntactic | calculate the similarity between terms then between their context using Cortical.io | 28 data sets | R: 0.25; P: 0.51 |
| Parra et al. 2015 [118] | General Assessment | a set of reqs | Good or Bad Requ | lexical and syntactic features | ML | 24 quality metrics that reflect lexical and syntactic information | C4.5*, PART, with Bagging and Boosting | INCOSE corpus | Acc: 0.87 |
| Femmer et al. 2017 [102] | General Assessment | a set of Reqs | quality scores | lexical and syntactic features | rule based | POS Tagging; Lemmatization | a set of rules and heuristics | 1471 reqs | P: 0.59; R:0.82 |
| Ferrari et al. 2018 [103] | General Assessment | a Requirement | 10 types of defects | lexical and syntactic features | rule based | Tokenization; POS Tagging; Shallow Parsing; Gazetters; JAPE rules; | dictionaties and rules | D-Pilot and D-Large | P: 0.83; R:0.85 |
| Ferrari et al. 2014 [111] | Incompleteness Detection | a set of Reqs | Completeness score | lexical and syntactic features | rule based | POS Tagging, extract POS-based patterns (adjective+noun) | calculate completeness score based on C-NC Values | pilot study | - |

**TABLE 5.** ***(Continued.)*** **The Detailed Info for all 104 selected papers.**

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Liu et al. 2021 [112] | Incompleteness Detection | SRS Document | is incomplete? | advanced embedding | ML | Tokenization, POS, Stop words removal; Lemmatization; skip-thoughts sent2vec encoder | a graph-based clustering | a set of aerospace reqs | F1:0.52 |
| Baumer et al. 2018 [113] | Incompleteness and Ambiguity Detection | a Req | solve predefined defect cases | ontology-based | rule based | Semantic Labeling; Stopwords removal; POS tagging; WordNet | a set of predefined rules to solve some defect cases | 400 selected requirements | F1: Incomp.(0.76); Ambig. (0.78) |
| Cruz et al. 2017 [117] | Vagueness Detection | a Req | is vague? | lexical and syntactic features | rule based | sentence splitrer, POS-tagging, get adjectives and adverbs that match a black-list of known vague terms or does not match a whitelist of not vague terms. | pilot study | P: 0.34; R:0.94 | |
| Arora et al. 2015 [114] | conformance with templates | a Req | Conformance or Not | lexical and syntactic features | rule based | Tokenization; POS tagging; NER; Chunking | Req Exp based on pattern matching | 4 test cases | F2: 0.96-1.0 |
| Abualhaija et al. 2020 [96] | Extraction from Docs | a textual document | Req or Not Req | lexical and syntactic features | ML | 20 Features: token-based features, syntactic features, semantic features, frequency-based features | RF with cost sensitive learning + a set of Post-Processing steps | 33 industrial set of reqs. | P: 0.81, R: 0.95 |
| Haris et al. 2020 [95] | Extraction from Docs | SRS | Req or Not Req | lexical and syntactic features | rule based | POS Tagging, chunking | POS tagging pattern | PURE dataset | P:0.64-1.0; R: 0.64-0.89 |
| Quirchmayr et al. 2018 [94] | Extraction from Docs | user manual | list of features | lexical and syntactic features | rule based | POS tagging; terms extraction ; parsing; Pattern-based parse tree transformations and correction | Syntactical patterns | industrial case | F1:0.92 |
| Wang et al. 2013 [93] | Extraction from Docs | a textual document | requirements | lexical and syntactic features | ML | Tokenization, lemmatization, pos tagging, dependency parsing | Bi-LSTM-CRF | 22 SRS documents | F1: 0.86 |
| Shi et al. 2021 [92] | Extraction from Docs | Mailing list | Feature requests | lexical and syntactic features | rule based | 81 fuzzy rules | 10 semantic sequence patterns | 317 emails from Ubuntu community | Avg precision: 0.76 Avg recall: 0.86 |
| De et al. 2021 [84] | Extraction from Reviews | user review | extract chunk which represent a feature | advanced embedding | ML | tokenization, BERT | use B,I,O annotations to label boundaries | reviews dataset for 8 Apps (125 for each) | F1 Exact Match (0.46); Partial Match (0.62) |
| Bakar et al. 2016 [85] | Extraction from Reviews | user review | features | Topic modeling | rule based | stop words removal, lemmatization, pos tagging, TF-IDF, Term-Document Matrix, | syntactic patterns to extract features | 32 software / 1253 sentences | P: 0.87 (0.62 avg); R: 0.86 (0.82 avg) |
| Carreno et al. 2013 [86] | Extraction from Reviews | user feedback | main topics | Topic modeling | ML | tokenization, stop words removal, | topic modeling and sentiment analysis | data set of 327 comments | P: 0.90; R: 0.5-0.8 |
| Jha et al. 2018 [87] | Extraction from Reviews | user review | bugs or features | VSM | ML | (1) BOF: based a probabilistic frame semantic parser. (2) BOW: Stemmer, BOW | SVM*, NB | Three datasets of reviews | F1: Features (0.75); Bugs (0.85) |
| Li et al. 2018 [88] | Extraction from Reviews | user requests | classify user requests to 8 types | lexical and syntactic features | ML | Unigram; TF-IDE; lexical and syntactic patterns | SVM*, KNN, NB | KeePass, Mumble, WinMerge | Avg Acc: 0.77; Avg R: 0.54-0.81; Avg P:0.73-0.92 |
| Lu et al. 2017 [63] | Extraction from Reviews | user review | 4 reviews classes. | advanced embedding | ML | the paper used word2vec to produce: AUR-BoW | Naive Bayes, J48, and Bagging* | 4000 sentences | F1: 0.71 |
| Peng et al. 2016 [89] | Extraction from Reviews | user reviews | feature requests | lexical and syntactic features | ML | lexical processing, pos tagging, dependency parser | Naive Bayes | 1924 reviews | F: 0.82 |
| Yang et al. 2021 [90] | Extraction from Reviews | user reviews | features | lexical and syntactic features | ML | Sentence splitting, sentiment analysis, POS tagging, stemming | RF | reviews for 6 products | F1: 0.72 |
| Wu et al. 2021 [91] | Extraction from Reviews | reviews | key features | advanced embedding | ML | tokenization, pos tagging, dependency parsing, BERT | A regression model to identify features | 200 app with 1,1M reviews. | F1: 0.78 |
| Jiang et al. 2019 [99] | Extraction From Similar Apps | featurs+repo. of similar applications | suggested new features | Topic modeling | ML | BOW+LDA | clustering technique | 533 annotated features from 100 apps | Hit@15 score: up to 78% |
| Abbas et al. 2020 [100] | Extraction From Similar Apps | corpus of reqs | recommended requirement | VSM | ML | Stop words removal, pos tagger, Lemmatization, TF-IDF | clustering requirements+ use Neighbours to recommend reqs | 188 requirements | Acc: 0.74 |
| Do et al. 2020 [101] | Extraction From Similar Apps | set of requirements | suggested new requirements | advanced embedding | ML | stemming, tokenization, doc2vec, POS tagging | clustering based BIRCH algorithm | 571 systems from 3 domains | Human evaluation |
| Hussain et al. 2013 [138] | Effort Estimation | a set of reqs | Function Point Level | lexical and syntactic features | ML | features including: NP; Parentheses; Active Verbs; Tokens in parentheses; Conjunctions; Pronouns; VP; Words; Sentences; Uniques (hapax legomena) | C4.5 | 4 projects | F1: 0.66 |
| Choetkiertikul et al. 2018 [137] | Effort Estimation | requirements | story points | advanced embedding | ML | word embedding | LSTM and recurrent highway network | 23,313 issues | outperforms 3 common baselines |
| Ali et al. 2021 [132] | Prioritization | set of requirements | priority | VSM | rule based | Tokenization, POS tagging, Stemming, Stop words removal,BOW | case-based reasoning (CBR) technique | four different case studies | |
| Kifetew et al. 2021 [133] | Prioritization | set of reqs + reqs' feedbacks | priority | lexical and syntactic features | rule based | tokenization, pos tagging, stemming, sentiment analysis, intention score, severity score | extracts quantifiable properties relevant for prioritizing reqs. | two empirical studies | |
| McZara et al. 2015 [134] | Prioritization | set of requirements | priority | lexical and syntactic features | rule based | tokenization, stemming, pos tagging | SMT solver + AHP | 100 reqs (top 20 are annotated) | |
| Misra et al. 2014 [135] | Prioritization | set of requirements | priority | Topic modeling | rule based | pos tagging, entities and action extraction, LSA | similarity-based approach | 41 requirements | |
| Shafiq et al. 2021 [136] | Prioritization | a set of Reqs | suggested priority | lexical and syntactic features | ML | | semi-automatic approach | 19 projects | average top3 acc 81% |
| Al-Hroob et al. 2018 [122] | Modeling (UML) | a set of Reqs | Use Cases | lexical and syntactic features | ML | Tokenization, pos tagging, dependency parsing | ANN to predict the role of each word | 5 test cases | F1: 0.43-0.44 |
| Elallaoui et al. 2018 [119] | Modeling (UML) | a set of Reqs | use cases | lexical and syntactic features | rule based | pos tagging | rule based | 90 user stories | P: 0.87; R:0.85 |
| Hamza et al. 2019 [121] | Modeling (UML) | a set of Reqs | use cases | lexical and syntactic features | rule based | Tokenization; POS tagging; Chunking; Grammar patterns tagging | rule based | 4 test cases | P: 0.72; R: 0.69 |
| Tiwari et al. 2019 [120] | Modeling (UML) | a set of Reqs | use cases | lexical and syntactic features | rule based | POS Tagger, Dependency Tree | rule based | 10 case studies. | Human Evaluation |
| Haj et al. 2021 [130] | Modeling (SBVR) | a set of Reqs | SBVR | lexical and syntactic features | rule based | Lemmatization, pos tagging, dependency tree, NER, | rule based | 3 case studies | F1: 0.87 |

**TABLE 5.** *(Continued.)* **The Detailed Info for all 104 selected papers.**

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Lucassen et al. 2017 [126] | Modeling (Conceptual Model) | a set of Reqs | concepts and relations | lexical and syntactic features | rule based | tokenization; pos tagging, dependency parsing | a set of heuristic rules | 4 industrial data sets | R:0.88-0.97; P:0.92-0.98 |
| Thakur et al. 2016 [125] | Modeling (Conceptual Model) | a set of Reqs | concepts and relations | lexical and syntactic features | rule based | tokenization; pos tagging, dependency parsing | rule based | 40 use cases | Acc: 0.91 |
| Hamza et al. 2015 [124] | Modeling (Features Model) | a set of Reqs | Features Model | lexical and syntactic features | rule based | tokenization, pos tagging | a set of heuristic rules | 4 case studies | human evaluation |
| Sree-Kumar et al. 2018 [123] | Modeling (Features Model) | a set of Reqs | Features | lexical and syntactic features | rule based | Stop words removal, Tokenization, NP chunking, TF-IDF | Set of heuristics rules | 6 case studies | Features: P 0.40-0.73 / R 0.57-0.93 Relation: P 0.41-0.87 / R 0.48-0.76 |
| Arora et al. 2016 [127] | Glossary Terms Extraction | a set of Reqs | Glossary + clustering terms | ontology-based | rule based | tokenization, pos tagging, chunking, Soft TF-IDF, wordNet | a set of huristics + similarity-based approach | 3 sets each consists of 110-380 reqs | F: 0.52 - 0.64 |
| Bhatia et al. 2020 [128] | Glossary Terms Extraction | a set of Reqs | Glossary + clustering terms | advanced embedding | ML | Stop words removal, Tokenization, chunking; Lemmatization, WordNet to remove common concepts | Similarity Matrix is calculated based on fastText + EM for clustering | CrowdRE Dataset | F1: Terms (0.39); Clustering (0.65) |
| Gunes et al. 2020 [129] | Modeling (Goal Model) | a set of Reqs | goal model | ontology-based | rule based | stopwords removal, pos tagging, extract user story elements, similarity scores | huristic rules | a test case | - |
| Mishra et al. 2021 [139] | Language Model | corpus of texts from wiki | SE word embeddings model | advanced embedding | ML | word embedding | word2vec | size of our trained model is 92MB | - |
| Dollmann et al. 2016 [97] | Extraction From Docs + Semantic Labeling | a set of sentences | reqs+Semantic labeling | lexical and syntactic features | ML | Features: (1) Extraction Approach (BOW; Length; dependencies; POS tags). Semantic Labeling (Orthographic; Semantic using WordNet; dependencies) | ExtraTreeClassifier | dataset from SourceForge | F1: Extraction(0.91); semantic labeling (0.73) |
| Sainani et al. 2020 [98] | Extraction From Docs / Classification | Contracts documents | Obligation or not | advanced embedding | ML | stop words, lemmatization, bigrams and trigrams, TF-IDF | BiLSTM | contracts document (250 pages, 1608 sentences) | F1: Extraction (0.93); Classification (0.60-0.96) |

(vi) There is a need for more flexible approaches in terms of the ability to handle requirements that do not follow (completely or partially) standard requirements templates, which is the common case in real-life projects.

(vii) The number of researches that use NLP-based representation is still limited for some RE tasks (requirements prioritization, effort estimation, semantic-based quality assessment).

Finally, tables 4 and 5 summarize the extracted information for all 104 selected papers, in addition to their categorization based on their targeted RE task and the used text representation. In the matrix represented in Table 4, each cell includes the papers related to each category of RE tasks and text representation. Note that two papers ( [97] and [98]) are mentioned twice in the table since they are related to both requirement analysis and requirement extraction categories. Table 5 shows a detailed summary of the 104 papers. This table includes detailed information about the input and the output of the approach proposed in each paper. In addition, it encloses other descriptive information including the employed text processing steps, details about the solution technique (e.g., which ML approach is used), the used dataset, and a summary of the results. This table provides a starting point for researchers and practitioners in this field to obtain a quick overview of the state-of-the-art in each RE task, including the recommended dataset to be used and the most related publications.

## VII. THREATS TO VALIDITY

Likewise any secondary research process, it is almost impossible to guarantee that we found the entire population of all the relevant papers. However, several actions were undertaken to minimize threats to validity.

- To ensure the inclusion of almost all relevant academic works in the field, we followed a systematic mapping review methodology based on the recommenced guidelines for similar cases. [15].
- Five reputable and well-known data sources (Scopus, IEEE Xplore, ACM Digital Library, ScienceDirect, and SpringerLink) were chosen to maximize the number of candidate papers.
- We tried to make our search string as general as possible by including various synonyms for each term and by including papers that cover either syntactic or semantic aspects. However, the final search string may not encompass all the existing synonyms, which might lead to not capturing all the relevant studies. We mitigated this threat by checking the references of the final selected papers to add any additional relevant works.
- To minimize mistakes caused by subjective analyses, we followed a rigorous study selection process, guided by clear inclusion and exclusion criteria. However, the exclusion of papers published in languages other than English may have failed to potentially find some relevant works.
- When there were doubts or conflicts about whether to include an article or not, the final decision is discussed between authors.
- To obtain data consistency and avoid bias in data extraction, we defined a clear data extraction template and discussed our results in several brainstorming sessions.

## VIII. CONCLUSION

This study presents a systematic mapping review of the used NLP-based representations in various RE tasks. Starting from 2,227 papers retrieved from five well-known digital libraries, we recognized 104 primary papers fulfilling the inclusion and exclusion criteria. We analyzed these works and categorized them based on: the targeted RE task and the used text representation.

Our results indicate that about two-thirds of retrieved publications handle tasks related to requirements classification, requirements traceability, ambiguity detection, and extracting requirements from reviews and documents. On the other hand, Lexical and syntactic features are widely used to represent requirements (more than 45% of publications). Besides, a growing number of papers use advanced word embedding techniques, especially in the last two years. Moreover, we summarize the main research directions to represent requirements in each category, and identify the gaps and possible future directions.

The data extracted from the selected 104 papers and their categorization, can help as a useful set of references for further analysis in each task or category of tasks in RE. Besides, trends and gaps identified from this mapping study have provided many new ideas for research opportunities.

## REFERENCES

[1] A. Tahir and R. Ahmad, ''Requirement engineering practices—An empirical study,'' in *Proc. Int. Conf. Comput. Intell. Softw. Eng.*, Dec. 2010, pp. 1–5.

[2] *IEEE Standard Glossary of Software Engineering Terminology*, IEEE Standard 610.12-1990, 1990.

[3] T. Ambreen, N. Ikram, M. Usman, and M. Niazi, ''Empirical research in requirements engineering: Trends and opportunities,'' *Requirements Eng.*, vol. 23, no. 1, pp. 63–95, Mar. 2018.

[4] I. Jureta, J. Mylopoulos, and S. Faulkner, ''Revisiting the core ontology and problem in requirements engineering,'' in *Proc. 16th IEEE Int. Requirements Eng. Conf.*, Sep. 2008, pp. 71–80.

[5] A. Aurum and C. Wohlin, ''Requirements engineering: Setting the context,'' in *Engineering and Managing Software Requirements*. Berlin, Germany: Springer, 2005, pp. 1–15.

[6] A. van Lamsweerde, R. Darimont, and E. Letier, ''Managing conflicts in goal-driven requirements engineering,'' *IEEE Trans. Softw. Eng.*, vol. 24, no. 11, pp. 908–926, Nov. 1998.

[7] C. Denger, D. M. Berry, and E. Kamsties, ''Higher quality requirements specifications through natural language patterns,'' in *Proc. Symp. Secur. Privacy*, Nov. 2003, pp. 80–90.

[8] R. Vlas and W. N. Robinson, ''A rule-based natural language technique for requirements discovery and classification in open-source software development projects,'' in *Proc. 44th Hawaii Int. Conf. Syst. Sci.*, Jan. 2011, pp. 1–10.

[9] Z. Kurtanovic and W. Maalej, ''Automatically classifying functional and non-functional requirements using supervised machine learning,'' in *Proc. IEEE 25th Int. Requirements Eng. Conf. (RE)*, Sep. 2017, pp. 490–495.

[10] F. Dalpiaz, D. Dell'Anna, F. B. Aydemir, and S. Cevikol, ''Requirements classification with interpretable machine learning and dependency parsing,'' in *Proc. IEEE 27th Int. Requirements Eng. Conf. (RE)*, Sep. 2019, pp. 142–152.

[11] T. Hey, J. Keim, A. Koziolek, and W. F. Tichy, ''NoRBERT: Transfer learning for requirements classification,'' in *Proc. IEEE 28th Int. Requirements Eng. Conf. (RE)*, Aug. 2020, pp. 169–179.

[12] A. Ferrari, F. Dell'Orletta, A. Esuli, V. Gervasi, and S. Gnesi, ''Natural language requirements processing: A 4D vision,'' *IEEE Softw.*, vol. 34, no. 6, pp. 28–35, Nov. 2017.

[13] R. Garigliano, D. Perini, and L. Mich, ''Which semantics for requirements engineering: From shallow to deep,'' in *Proc. REFSQ Workshops*, 2018, pp. 1–5.

[14] A. Booth, A. Sutton, M. Clowes, and M. M.-S. James, *Systematic Approaches to a Successful Literature Review*. London, U.K.: Sage Publicaations, 2021.

[15] K. Petersen, S. Vakkalanka, and L. Kuzniarz, ''Guidelines for conducting systematic mapping studies in software engineering: An update,'' *Inf. Softw. Technol.*, vol. 64, pp. 1–18, Aug. 2015.

[16] D. Jurafsky, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Upper Saddle River, NJ, USA: Prentice-Hall, 2000.

[17] V. Teller, ''Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition,'' *Comput. Linguistics*, vol. 26, no. 4, pp. 638–641, 2000.

[18] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Upper Saddle River, NJ, USA: Prentice-Hall, 2000.

[19] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer, ''Feature-rich part-of-speech tagging with a cyclic dependency network,'' in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics Hum. Lang. Technol. (NAACL)*, 2003, pp. 252–259.

[20] S. Abney, ''Partial parsing via finite-state cascades,'' *Natural Lang. Eng.*, vol. 2, no. 4, pp. 337–344, Dec. 1996.

[21] M.-C. De Marneffe and C. D. Manning, ''Stanford typed dependencies manual,'' Stanford Univ., Stanford, CA, USA, Tech. Rep., 2008. [Online]. Available: http://nlp.stanford.edu/software/dependencies_manual.pdf

[22] D. Nadeau and S. Sekine, ''A survey of named entity recognition and classification,'' *Lingvisticae Investigationes*, vol. 30, no. 1, pp. 3–26, Aug. 2007.

[23] T. R. Gruber, ''A translation approach to portable ontology specifications,'' *Knowl. Acquisition*, vol. 5, no. 2, pp. 199–220, Jun. 1993.

[24] G. A. Miller, ''WordNet: A lexical database for English,'' *Commun. ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[25] Z. Wu and M. Palmer, ''Verb semantics and lexical selection,'' in *Proc. 32nd Annu. Meeting Assoc. Comput. Linguistics*, 1994, pp. 133–138.

[26] R. Rada, H. Mili, E. Bicknell, and M. Blettner, ''Development and application of a metric on semantic nets,'' *IEEE Trans. Syst., Man, Cybern.*, vol. 19, no. 1, pp. 17–30, Jan./Feb. 1989.

[27] G. Salton, *Automatic Text Processing*. London, U.K.: Longman, 1988.

[28] D. M. Blei, A. Y. Ng, and M. I. Jordan, ''Latent Dirichlet allocation,'' *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Jan. 2003.

[29] S. T. Dumais, ''Latent semantic analysis,'' *Annu. Rev. Inf. Sci. Technol.*, vol. 38, no. 1, pp. 188–230, 2004.

[30] T. Mikolov, K. Chen, G. Corrado, and J. Dean, ''Efficient estimation of word representations in vector space,'' 2013, *arXiv:1301.3781*.

[31] J. Pennington, R. Socher, and C. Manning, ''GloVe: Global vectors for word representation,'' in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1532–1543.

[32] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, ''BERT: Pre-training of deep bidirectional transformers for language understanding,'' in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1, Minneapolis, MN, USA: Assoc. Comput. Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: https://www.aclweb.org/anthology/N19-1423

[33] L. Zhao, W. Alhoshan, A. Ferrari, K. J. Letsholo, M. A. Ajagbe, E.-V. Chioasca, and R. T. Batista-Navarro, ''Natural language processing for requirements engineering: A systematic mapping study,'' *ACM Comput. Surv.*, vol. 54, no. 3, pp. 1–41, Apr. 2022.

[34] I. K. Raharjana, D. Siahaan, and C. Fatichah, ''User stories and natural language processing: A systematic literature review,'' *IEEE Access*, vol. 9, pp. 53811–53826, 2021.

[35] A. R. Amna and G. Poels, ''Ambiguity in user stories: A systematic literature review,'' *Inf. Softw. Technol.*, vol. 145, May 2022, Art. no. 106824.

[36] D. Dermeval, J. Vilela, I. I. Bittencourt, J. Castro, S. Isotani, and P. Brito, ''A systematic review on the use of ontologies in requirements engineering,'' in *Proc. Brazilian Symp. Softw. Eng.*, Sep. 2014, pp. 1–10.

[37] F. Nazir, W. H. Butt, M. W. Anwar, and M. A. K. Khattak, ''The applications of natural language processing (NLP) for software requirement engineering—A systematic literature review,'' in *Proc. Int. Conf. Inf. Sci. Appl.* Singapore: Springer, 2017, pp. 485–493.

[38] F. Bozyiğit, Ö. Aktaş, and D. Kılınç, ''Linking software requirements and conceptual models: A systematic literature review,'' *Eng. Sci. Technol., Int. J.*, vol. 24, no. 1, pp. 71–82, Feb. 2021.

[39] C. Baker, L. Deng, S. Chakraborty, and J. Dehlinger, ''Automatic multiclass non-functional software requirements classification using neural networks,'' in *Proc. IEEE 43rd Annu. Comput. Softw. Appl. Conf. (COMPSAC)*, vol. 2, Jul. 2019, pp. 610–615.

[40] E. D. Canedo and B. Mendes, ''Software requirements classification using machine learning algorithms,'' *Entropy*, vol. 22, no. 9, p. 1057, Sep. 2020.

[41] A. Casamayor, D. Godoy, and M. Campo, "Identification of non-functional requirements in textual specifications: A semi-supervised learning approach," *Inf. Softw. Technol.*, vol. 52, no. 4, pp. 436–445, Apr. 2010.
[42] V. M. Khatian, Q. A. Arain, M. Alenezi, M. O. Raza, F. Shaikh, and I. Farah, "Comparative analysis for predicting non-functional requirements using supervised machine learning," in *Proc. 1st Int. Conf. Artif. Intell. Data Anal. (CAIDA)*, Apr. 2021, pp. 7–12.
[43] M. A. Rahman, M. A. Haque, M. N. A. Tawhid, and M. S. Siddik, "Classifying non-functional requirements using RNN variants for quality software development," in *Proc. 3rd ACM SIGSOFT Int. Workshop Mach. Learn. Techn. Softw. Quality Eval. (MaLTeSQuE)*, 2019, pp. 25–30.
[44] A. Rashwan, O. Ormandjieva, and R. Witte, "Ontology-based classification of non-functional requirements in software specifications: A new corpus and SVM-based classifier," in *Proc. IEEE 37th Annu. Comput. Softw. Appl. Conf.*, Jul. 2013, pp. 381–386.
[45] M. Younas, D. N. A. Jawawi, I. Ghani, and M. A. Shah, "Extraction of non-functional requirement using semantic similarity distance," *Neural Comput. Appl.*, vol. 32, no. 11, pp. 7383–7397, Jun. 2020.
[46] A. Mahmoud and G. Williams, "Detecting, classifying, and tracing non-functional software requirements," *Requirements Eng.*, vol. 21, no. 3, pp. 357–381, Sep. 2016.
[47] D. Kici, G. Malik, M. Cevik, D. Parikh, and A. Başar, "A BERT-based transfer learning approach to text classification on software requirements specifications," in *Proc. Can. Conf. Artif. Intell.*, Jun. 2021, pp. 1–13.
[48] I. M. S. Raharja and D. O. Siahaan, "Classification of non-functional requirements using fuzzy similarity KNN based on ISO/IEC 25010," in *Proc. 12th Int. Conf. Inf. Commun. Technol. Syst. (ICTS)*, Jul. 2019, pp. 264–269.
[49] R. Krasniqi and A. Agrawal, "Analyzing and detecting emerging quality-related concerns across OSS defect report summaries," in *Proc. IEEE Int. Conf. Softw. Anal., Evol. Reeng. (SANER)*, Mar. 2021, pp. 12–23.
[50] D. N. Palacio, D. McCrystal, K. Moran, C. Bernal-Cardenas, D. Poshyvanyk, and C. Shenefiel, "Learning to identify security-related issues using convolutional neural networks," in *Proc. IEEE Int. Conf. Softw. Maintenance Evol. (ICSME)*, Sep. 2019, pp. 140–144.
[51] A. Kobilica, M. Ayub, and J. Hassine, "Automated identification of security requirements: A machine learning approach," in *Proc. Eval. Assessment Softw. Eng.*, Apr. 2020, pp. 475–480.
[52] T. Li and Z. Chen, "An ontology-based learning approach for automatically classifying security requirements," *J. Syst. Softw.*, vol. 165, Jul. 2020, Art. no. 110566.
[53] D. Ott, "Automatic requirement categorization of large natural language specifications at mercedes-benz for review improvements," in *Proc. Int. Work. Conf. Requirements Eng., Found. Softw. Quality*. Berlin, Germany: Springer, 2013, pp. 50–64.
[54] R. Chatterjee, A. Ahmed, and P. R. Anish, "Identification and classification of architecturally significant functional requirements," in *Proc. IEEE 7th Int. Workshop Artif. Intell. Requirements Eng. (AIRE)*, Sep. 2020, pp. 9–17.
[55] H. Wang, G. Shen, Z. Huang, Y. Yu, and K. Chen, "Analyzing close relations between target artifacts for improving IR-based requirement traceability recovery," *Frontiers Inf. Technol. Electron. Eng.*, vol. 22, no. 7, pp. 957–968, Jul. 2021.
[56] S. Das, N. Deb, A. Cortesi, and N. Chaki, "Sentence embedding models for similarity detection of software requirements," *Social Netw. Comput. Sci.*, vol. 2, no. 2, pp. 1–11, Apr. 2021.
[57] M. B. Blake, I. Saleh, Y. Wei, I. D. Schlesinger, A. Yale-Loehr, and X. Liu, "Shared service recommendations from requirement specifications: A hybrid syntactic and semantic toolkit," *Inf. Softw. Technol.*, vol. 57, pp. 392–404, Jan. 2015.
[58] R. Sonbol, G. Rebdawi, and N. Ghneim, "Towards a semantic representation for functional software requirements," in *Proc. IEEE 7th Int. Workshop Artif. Intell. Requirements Eng. (AIRE)*, Sep. 2020, pp. 1–8.
[59] W. Alhoshan, R. Batista-Navarro, and L. Zhao, "Using frame embeddings to identify semantically related software requirements," in *Proc. REFSQ Workshops*, 2019, pp. 1–9.
[60] G. Deshpande, C. Arora, and G. Ruhe, "Data-driven elicitation and optimization of dependencies between requirements," in *Proc. IEEE 27th Int. Requirements Eng. Conf. (RE)*, Sep. 2019, pp. 416–421.
[61] R. Samer, M. Stettinger, M. Atas, A. Felfernig, G. Ruhe, and G. Deshpande, "New approaches to the identification of dependencies between requirements," in *Proc. IEEE 31st Int. Conf. Tools Artif. Intell. (ICTAI)*, Nov. 2019, pp. 1265–1270.
[62] H. Sultanov, J. H. Hayes, and W.-K. Kong, "Application of swarm techniques to requirements tracing," *Requirements Eng.*, vol. 16, no. 3, pp. 209–226, Sep. 2011.
[63] M. Lu and P. Liang, "Automatic classification of non-functional requirements from augmented app user reviews," in *Proc. 21st Int. Conf. Eval. Assessment Softw. Eng.*, Jun. 2017, pp. 344–353.
[64] G. Deshpande, B. Sheikhi, S. Chakka, D. L. Zotegouon, M. N. Masahati, and G. Ruhe, "Is BERT the new silver bullet?—An empirical investigation of requirements dependency classification," in *Proc. IEEE 29th Int. Requirements Eng. Conf. Workshops (REW)*, Sep. 2021, pp. 136–145.
[65] M. Aldekhail and M. Almasri, "Intelligent identification and resolution of software requirement conflicts: Assessment and evaluation," *Comput. Syst. Sci. Eng.*, vol. 40, no. 2, pp. 469–489, 2022.
[66] A. Nicholson and G. Jin, "Issue link label recovery and prediction for open source software," in *Proc. IEEE 29th Int. Requirements Eng. Conf. Workshops (REW)*, Sep. 2021, pp. 126–135.
[67] J. Guo, J. Cheng, and J. Cleland-Huang, "Semantically enhanced software traceability using deep learning techniques," in *Proc. IEEE/ACM 39th Int. Conf. Softw. Eng. (ICSE)*, May 2017, pp. 3–14.
[68] B. Wang, R. Peng, Z. Wang, X. Wang, and Y. Li, "An automated hybrid approach for generating requirements trace links," *Int. J. Softw. Eng. Knowl. Eng.*, vol. 30, no. 7, pp. 1005–1048, Jul. 2020.
[69] M. H. Sadreddini, S. M. Fakhrahmad, and A. H. Rasekh, "Mining traces between source code and textual documents," *Int. J. Comput. Appl. Technol.*, vol. 59, no. 1, p. 43, 2019.
[70] A. Mahmoud and N. Niu, "On the role of semantics in automated requirements tracing," *Requirements Eng.*, vol. 20, no. 3, pp. 281–300, Sep. 2015.
[71] N. Ali, H. Cai, A. Hamou-Lhadj, and J. Hassine, "Exploiting parts-of-speech for effective automated requirements traceability," *Inf. Softw. Technol.*, vol. 106, pp. 126–141, Feb. 2019.
[72] V. Leitão and I. Medeiros, "SRXCRM: Discovering association rules between system requirements and product specifications," in *Proc. REFSQ Workshops*, 2021, pp. 1–9.
[73] J. K. Chhabra, "Requirements traceability through information retrieval using dynamic integration of structural and co-change coupling," in *Proc. Int. Conf. Adv. Inform. Comput. Res.* Singapore: Springer, 2017, pp. 107–118.
[74] C. Arora, M. Sabetzadeh, A. Goknil, L. C. Briand, and F. Zimmer, "Change impact analysis for natural language requirements: An NLP approach," in *Proc. IEEE 23rd Int. Requirements Eng. Conf. (RE)*, Aug. 2015, pp. 6–15.
[75] T. N. Al-Otaiby, M. AlSherif, and W. P. Bond, "Toward software requirements modularization using hierarchical clustering techniques," in *Proc. 43rd Annu. Southeast Regional Conf.*, 2005, pp. 223–228.
[76] K. J. Gulle, N. Ford, P. Ebel, F. Brokhausen, and A. Vogelsang, "Topic modeling on user stories using word Mover's distance," in *Proc. IEEE 7th Int. Workshop Artif. Intell. Requirements Eng. (AIRE)*, Sep. 2020, pp. 52–60.
[77] A. Casamayor, D. Godoy, and M. Campo, "Functional grouping of natural language requirements for assistance in architectural software design," *Knowl.-Based Syst.*, vol. 30, pp. 78–86, Jun. 2012.
[78] J. Misra, S. Sengupta, and S. Podder, "Topic cohesion preserving requirements clustering," in *Proc. EEE/ACM 5th Int. Workshop Realizing Artif. Intell. Synergies Softw. Eng. (RAISE)*, May 2016, pp. 22–28.
[79] H. E. Salman, M. Hammad, A.-D. Seriai, and A. Al-Sbou, "Semantic clustering of functional requirements using agglomerative hierarchical clustering," *Information*, vol. 9, no. 9, p. 222, Sep. 2018.
[80] M. Mezghani, J. Kang, and F. Sedes, "Industrial requirements classification for redundancy and inconsistency detection in SEMIOS," in *Proc. IEEE 26th Int. Requirements Eng. Conf. (RE)*, Aug. 2018, pp. 297–303.
[81] Y. Wang, "Semantic information extraction for software requirements using semantic role labeling," in *Proc. IEEE Int. Conf. Prog. Informat. Comput. (PIC)*, Dec. 2015, pp. 332–337.
[82] T. Diamantopoulos, M. Roth, A. Symeonidis, and E. Klein, "Software requirements as an application domain for natural language processing," *Lang. Resour. Eval.*, vol. 51, no. 2, pp. 495–524, Jun. 2017.
[83] Y. Wang, "Automatic semantic analysis of software requirements through machine learning and ontology approach," *J. Shanghai Jiaotong Univ., Sci.*, vol. 21, no. 6, pp. 692–701, Dec. 2016.
[84] A. F. de Araújo and R. M. Marcacini, "RE-BERT: Automatic extraction of software requirements from app reviews using BERT language model," in *Proc. 36th Annu. ACM Symp. Appl. Comput.*, Mar. 2021, pp. 1321–1327.

[85] N. H. Bakar, Z. M. Kasirun, N. Salleh, and H. A. Jalab, "Extracting features from online software reviews to aid requirements reuse," *Appl. Soft Comput.*, vol. 49, pp. 1297–1315, Dec. 2016.

[86] L. V. G. Carreno and K. Winbladh, "Analysis of user comments: An approach for software requirements evolution," in *Proc. 35th Int. Conf. Softw. Eng. (ICSE)*, May 2013, pp. 582–591.

[87] N. Jha and A. Mahmoud, "Using frame semantics for classifying and summarizing application store reviews," *Empirical Softw. Eng.*, vol. 23, no. 6, pp. 3734–3767, Dec. 2018.

[88] C. Li, L. Huang, J. Ge, B. Luo, and V. Ng, "Automatically classifying user requests in crowdsourcing requirements engineering," *J. Syst. Softw.*, vol. 138, pp. 108–123, Apr. 2018.

[89] Z. Peng, J. Wang, K. He, and M. Tang, "An approach of extracting feature requests from app reviews," in *Proc. Int. Conf. Collaborative Comput., Netw., Appl. Worksharing*. Cham, Switzerland: Springer, 2016, pp. 312–323.

[90] C. Yang, L. Wu, C. Yu, and Y. Zhou, "A phrase-level user requests mining approach in mobile application reviews: Concept, framework, and operation," *Information*, vol. 12, no. 5, p. 177, Apr. 2021.

[91] H. Wu, W. Deng, X. Niu, and C. Nie, "Identifying key features from app user reviews," in *Proc. IEEE/ACM 43rd Int. Conf. Softw. Eng. (ICSE)*, May 2021, pp. 922–932.

[92] L. Shi, C. Chen, Q. Wang, and B. Boehm, "Automatically detecting feature requests from development emails by leveraging semantic sequence mining," *Requirements Eng.*, vol. 26, no. 2, pp. 255–271, Jun. 2021.

[93] Y. Wang, I. L. M. Gutiérrez, K. Winbladh, and H. Fang, "Automatic detection of ambiguous terminology for software requirements," in *Proc. Int. Conf. Appl. Natural Lang. Inf. Syst.* Berlin, Germany: Springer, 2013, pp. 25–37.

[94] T. Quirchmayr, B. Paech, R. Kohl, H. Karey, and G. Kasdepke, "Semi-automatic rule-based domain terminology and software feature-relevant information extraction from natural language user manuals," *Empirical Softw. Eng.*, vol. 23, no. 6, pp. 3630–3683, Dec. 2018.

[95] M. S. Haris, T. A. Kurniawan, and F. Ramdani, "Automated features extraction from software requirements specification (SRS) documents as the basis of software product line (SPL) engineering," *J. Inf. Technol. Comput. Sci.*, vol. 5, no. 3, pp. 279–292, Dec. 2020.

[96] S. Abualhaija, C. Arora, M. Sabetzadeh, L. C. Briand, and M. Traynor, "Automated demarcation of requirements in textual specifications: A machine learning-based approach," *Empirical Softw. Eng.*, vol. 25, no. 6, pp. 5454–5497, Nov. 2020.

[97] M. Dollmann and M. Geierhos, "On- and off-topic classification and semantic annotation of user-generated software requirements," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 1807–1816.

[98] A. Sainani, P. R. Anish, V. Joshi, and S. Ghaisas, "Extracting and classifying requirements from software engineering contracts," in *Proc. IEEE 28th Int. Requirements Eng. Conf. (RE)*, Aug. 2020, pp. 147–157.

[99] H. Jiang, J. Zhang, X. Li, Z. Ren, D. Lo, X. Wu, and Z. Luo, "Recommending new features from mobile app descriptions," *ACM Trans. Softw. Eng. Methodol.*, vol. 28, no. 4, pp. 1–29, Oct. 2019.

[100] M. Abbas, M. Saadatmand, E. Enoiu, D. Sundamark, and C. Lindskog, "Automated reuse recommendation of product line assets based on natural language requirements," in *Proc. Int. Conf. Softw. Softw. Reuse*. Cham, Switzerland: Springer, 2020, pp. 173–189.

[101] Q. A. Do, T. Bhowmik, and G. L. Bradshaw, "Capturing creative requirements via requirements reuse: A machine learning-based approach," *J. Syst. Softw.*, vol. 170, Dec. 2020, Art. no. 110730.

[102] H. Femmer, D. M. Fernández, S. Wagner, and S. Eder, "Rapid quality assurance with requirements smells," *J. Syst. Softw.*, vol. 123, pp. 190–213, Jan. 2017.

[103] A. Ferrari, G. Gori, B. Rosadini, I. Trotta, S. Bacherini, A. Fantechi, and S. Gnesi, "Detecting requirements defects with NLP patterns: An industrial experience in the railway domain," *Empirical Softw. Eng.*, vol. 23, no. 6, pp. 3684–3733, Dec. 2018.

[104] S. Mishra and A. Sharma, "On the use of word embeddings for identifying domain specific ambiguities in requirements," in *Proc. IEEE 27th Int. Requirements Eng. Conf. Workshops (REW)*, Sep. 2019, pp. 234–240.

[105] J. Matsuoka and Y. Lepage, "Ambiguity spotting using WordNet semantic similarity in support to recommended practice for software requirements specifications," in *Proc. 7th Int. Conf. Natural Lang. Process. Knowl. Eng.*, Nov. 2011, pp. 479–484.

[106] A. Ferrari and A. Esuli, "An NLP approach for cross-domain ambiguity detection in requirements engineering," *Automated Softw. Eng.*, vol. 26, no. 3, pp. 559–598, Sep. 2019.

[107] J. Misra and S. Das, "Entity disambiguation in natural language text requirements," in *Proc. 20th Asia–Pacific Softw. Eng. Conf. (APSEC)*, vol. 1, Dec. 2013, pp. 239–246.

[108] M. Osama, A. Zaki-Ismail, M. Abdelrazek, J. Grundy, and A. Ibrahim, "Score-based automatic detection and resolution of syntactic ambiguity in natural language requirements," in *Proc. IEEE Int. Conf. Softw. Maintenance Evol. (ICSME)*, Sep. 2020, pp. 651–661.

[109] S. Ezzini, S. Abualhaija, C. Arora, M. Sabetzadeh, and L. C. Briand, "Using domain-specific corpora for improved handling of ambiguity in requirements," in *Proc. IEEE/ACM 43rd Int. Conf. Softw. Eng. (ICSE)*, May 2021, pp. 1485–1497.

[110] A. Ferrari and S. Gnesi, "Using collective intelligence to detect pragmatic ambiguities," in *Proc. 20th IEEE Int. Requirements Eng. Conf. (RE)*, Sep. 2012, pp. 191–200.

[111] A. Ferrari, F. Dell'Orletta, G. O. Spagnolo, and S. Gnesi, "Measuring and improving the completeness of natural language requirements," in *Proc. Int. Work. Conf. Requirements Eng., Found. Softw. Quality*. Cham, Switzerland: Springer, 2014, pp. 23–38.

[112] C. Liu, Z. Zhao, L. Zhang, and Z. Li, "Automated conditional statements checking for complete natural language requirements specification," *Appl. Sci.*, vol. 11, no. 17, p. 7892, Aug. 2021.

[113] F. S. Bäumer and M. Geierhos, "Flexible ambiguity resolution and incompleteness detection in requirements descriptions via an indicator-based configuration of text analysis pipelines," in *Proc. 51st Hawaii Int. Conf. Syst. Sci.*, 2018, pp. 1–10.

[114] C. Arora, M. Sabetzadeh, L. Briand, and F. Zimmer, "Automated checking of conformance to requirements templates using natural language processing," *IEEE Trans. Softw. Eng.*, vol. 41, no. 10, pp. 944–968, Oct. 2015.

[115] C. Rupp and K. Pohl, *Requirements Engineering Fundamentals*. San Rafael, CA, USA: Rocky Nook, 2011.

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

[117] B. D. Cruz, B. Jayaraman, A. Dwarakanath, and C. Mcmillan, "Detecting vague words & phrases in requirements documents in a multilingual environment," in *Proc. IEEE 25th Int. Requirements Eng. Conf. (RE)*, Sep. 2017, pp. 233–242.

[118] E. Parra, C. Dimou, J. Llorens, V. Moreno, and A. Fraga, "A methodology for the classification of quality of requirements using machine learning techniques," *Inf. Softw. Technol.*, vol. 67, pp. 180–195, Nov. 2015.

[119] M. Elallaoui, K. Nafil, and R. Touahni, "Automatic transformation of user stories into UML use case diagrams using NLP techniques," *Proc. Comput. Sci.*, vol. 130, pp. 42–49, Jan. 2018.

[120] S. Tiwari, D. Ameta, and A. Banerjee, "An approach to identify use case scenarios from textual requirements specification," in *Proc. 12th Innov. Softw. Eng. Conf., Formerly Known India Softw. Eng. Conf.*, Feb. 2019, pp. 1–11.

[121] Z. A. Hamza and M. Hammad, "Generating UML use case models from software requirements using natural language processing," in *Proc. 8th Int. Conf. Modeling Simulation Appl. Optim. (ICMSAO)*, Apr. 2019, pp. 1–6.

[122] A. Al-Hroob, A. T. Imam, and R. Al-Heisa, "The use of artificial neural networks for extracting actions and actors from requirements document," *Inf. Softw. Technol.*, vol. 101, pp. 1–15, Sep. 2018.

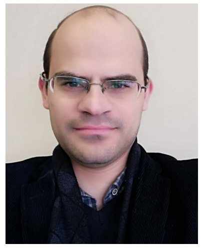
[123] A. Sree-Kumar, E. Planas, and R. Clarisó, "Extracting software product line feature models from natural language specifications," in *Proc. 22nd Int. Syst. Softw. Product Line Conf.*, vol. 1, Sep. 2018, pp. 43–53.

[124] M. Hamza and R. J. Walker, "Recommending features and feature relationships from requirements documents for software product lines," in *Proc. IEEE/ACM 4th Int. Workshop Realizing Artif. Intell. Synergies Softw. Eng.*, May 2015, pp. 25–31.

[125] J. S. Thakur and A. Gupta, "Identifying domain elements from textual specifications," in *Proc. 31st IEEE/ACM Int. Conf. Automated Softw. Eng.*, Aug. 2016, pp. 566–577.

[126] 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 Eng.*, vol. 22, no. 3, pp. 339–358, Sep. 2017.

[127] C. Arora, M. Sabetzadeh, L. Briand, and F. Zimmer, "Automated extraction and clustering of requirements glossary terms," *IEEE Trans. Softw. Eng.*, vol. 43, no. 10, pp. 918–945, Oct. 2017.
[128] K. Bhatia, S. Mishra, and A. Sharma, "Clustering glossary terms extracted from large-sized software requirements using FastText," in *Proc. 13th Innov. Softw. Eng. Conf., Formerly Known India Softw. Eng. Conf.*, Feb. 2020, pp. 1–11.
[129] T. Gunes and F. B. Aydemir, "Automated goal model extraction from user stories using NLP," in *Proc. IEEE 28th Int. Requirements Eng. Conf. (RE)*, Aug. 2020, pp. 382–387.
[130] A. Haj, A. Jarrar, Y. Balouki, and T. Gadir, "The semantic of business vocabulary and business rules: An automatic generation from textual statements," *IEEE Access*, vol. 9, pp. 56506–56522, 2021.
[131] P. Tzortzopoulos, R. Cooper, P. Chan, and M. Kagioglou, "Clients' activities at the design front-end," *Des. Stud.*, vol. 27, no. 6, pp. 657–683, 2006.
[132] S. Ali, Y. Hafeez, S. Hussain, S. Yang, and M. Jamal, "Requirement prioritization framework using case-based reasoning: A mining-based approach," *Expert Syst.*, vol. 38, no. 8, Dec. 2021, Art. no. e12770.
[133] F. M. Kifetew, A. Perini, A. Susi, A. Siena, D. Muñante, and I. Morales-Ramirez, "Automating user-feedback driven requirements prioritization," *Inf. Softw. Technol.*, vol. 138, Oct. 2021, Art. no. 106635.
[134] J. McZara, S. Sarkani, T. Holzer, and T. Eveleigh, "Software requirements prioritization and selection using linguistic tools and constraint solvers—A controlled experiment," *Empirical Softw. Eng.*, vol. 20, no. 6, pp. 1721–1761, Dec. 2015.
[135] J. Misra, S. Sengupta, and S. Das, "Latent semantic centrality based automated requirements prioritization," in *Proc. 7th India Softw. Eng. Conf. (ISEC)*, 2014, pp. 1–10.
[136] S. Shafiq, A. Mashkoor, C. Mayr-Dorn, and A. Egyed, "NLP4IP: Natural language processing-based recommendation approach for issues prioritization," in *Proc. 47th Euromicro Conf. Softw. Eng. Adv. Appl. (SEAA)*, Sep. 2021, pp. 99–108.
[137] M. Choetkiertikul, H. K. Dam, T. Tran, T. Pham, A. Ghose, and T. Menzies, "A deep learning model for estimating story points," *IEEE Trans. Softw. Eng.*, vol. 45, no. 7, pp. 637–656, Jul. 2019.
[138] I. Hussain, L. Kosseim, and O. Ormandjieva, "Approximation of COSMIC functional size to support early effort estimation in agile," *Data Knowl. Eng.*, vol. 85, pp. 2–14, May 2013.
[139] S. Mishra and A. Sharma, "Crawling Wikipedia pages to train word embeddings model for software engineering domain," in *Proc. 14th Innov. Softw. Eng. Conf., Formerly Known India Softw. Eng. Conf.*, Feb. 2021, pp. 1–5.
[140] U. Shah, S. Patel, and D. C. Jinwala, "Detecting intra-conflicts in non-functional requirements," *Int. J. Uncertainty, Fuzziness Knowl.-Based Syst.*, vol. 29, no. 3, pp. 435–461, Jun. 2021.
[141] V. Silva-Rodríguez, S. E. Nava-Muñoz, L. A. Castro, F. E. Martínez-Pérez, H. G. Pérez-González, and F. Torres-Reyes, "Classifying design-level requirements using machine learning for a recommender of interaction design patterns," *IET Softw.*, vol. 14, no. 5, pp. 544–552, Oct. 2020.
[142] C. F. Baker, C. J. Fillmore, and J. B. Lowe, "The Berkeley FrameNet project," in *Proc. 17th Int. Conf. Comput. Linguistics*, 1998, pp. 1–5.
[143] Y. Li and T. Yang, "Word embedding for understanding natural language: A survey," in *Guide to Big Data Applications*. Cham, Switzerland: Springer, 2018, pp. 83–104.
[144] F. Dalpiaz, I. van der Schalk, S. Brinkkemper, F. B. Aydemir, and G. Lucassen, "Detecting terminological ambiguity in user stories: Tool and experimentation," *Inf. Softw. Technol.*, vol. 110, pp. 3–16, Jun. 2019.
[145] C. De Boom, S. Van Canneyt, T. Demeester, and B. Dhoedt, "Representation learning for very short texts using weighted word embedding aggregation," *Pattern Recognit. Lett.*, vol. 80, pp. 150–156, Sep. 2016.
[146] S. Dupond, "A thorough review on the current advance of neural network structures," *Annu. Rev. Control*, vol. 14, pp. 200–230, Jan. 2019.
[147] Z. Liu, B. Li, J. Wang, and R. Yang, "Requirements engineering for crossover services: Issues, challenges and research directions," *IET Softw.*, vol. 15, no. 1, pp. 107–125, Feb. 2021.
[148] A. Mavin, P. Wilkinson, S. Teufl, H. Femmer, J. Eckhardt, and J. Mund, "Does goal-oriented requirements engineering achieve its goal?" in *Proc. IEEE 25th Int. Requirements Eng. Conf. (RE)*, Sep. 2017, pp. 174–183.
[149] U. Eklund, H. H. Olsson, and N. J. Strøm, "Industrial challenges of scaling agile in mass-produced embedded systems," in *Proc. Int. Conf. Agile Softw. Develop.* Cham, Switzerland: Springer, 2014, pp. 30–42.
[150] G. Fanmuy, A. Fraga, and J. Llorens, "Requirements verification in the industry," in *Complex Systems Design & Management*. Berlin, Germany: Springer, 2012, pp. 145–160.
[151] Y. Wautelet, S. Heng, M. Kolp, and I. Mirbel, "Unifying and extending user story models," in *Proc. Int. Conf. Adv. Inf. Syst. Eng.* Cham, Switzerland: Springer, 2014, pp. 211–225.
[152] A. D. Thommazo, T. Ribeiro, G. Olivatto, V. Werneck, and S. Fabbri, "An automatic approach to detect traceability links using fuzzy logic," in *Proc. 27th Brazilian Symp. Softw. Eng.*, Oct. 2013, pp. 21–30.
[153] Y. Wang and J. Zhang, "Experiment on automatic functional requirements analysis with the EFRF's semantic cases," in *Proc. Int. Conf. Prog. Informat. Comput. (PIC)*, Dec. 2016, pp. 636–642.



**RIAD SONBOL** received the B.S. and M.S. degrees in informatics engineering from the Higher Institute for Applied Sciences and Technology (HIAST), Syria, in 2008 and 2017, respectively. Currently, he is pursuing the Ph.D. degree in natural language processing. Since 2008, he has been an Assistant Teacher for a number of AI courses with HIAST. He has published more than ten research papers related to natural language processing and software engineering domain. His research interests include natural language processing, machine learning, requirement engineering, and business process management.

**GHAIDA REBDAWI** received the B.S. degree in informatics engineering from the Higher Institute for Applied Sciences and Technology, Syria, in 1985, and the DEA and Ph.D. degrees in applied automatic and informatics from the Institut National des Sciences Appliquées de Lyon, France, in 1987 and 1990, respectively. In 1991, she joined as a Research Assistant Professor at the Informatics Department, Higher Institute for Applied Sciences and Technology, where she has been a Research Professor, since 2012. She coauthored many academic books in computer science, and more than 20 articles. Her research interests include software engineering, requirements engineering, natural language processing, and business process management.

**NADA GHNEIM** received the Ph.D. degree (DEA) in artificial intelligence (image, robotics, vision) from the National High School of Computer Science and Applied Mathematics in Grenoble (ENSIMAG), France, in 1993, and the Ph.D. degree in language sciences (speech communication) from the Institut de la Communication Parlée, Stendhal (Grenoble III) University, France, in 1997. She is an Assistant Professor with the Faculty of Informatics Communication Engineering, Arab International University (AIU), Damascus, Syria. She is also a Researcher/Lecturer with the Higher Institute for Applied Sciences and Technology (HIAST) and the Information Technology Engineering Faculty, Damascus University. She has many publications in her research areas. Her research interests include speech and natural language processing domain, such as Arabic text-to-speech, sentiment analysis, morphological and syntactic analysis, dictionary, and ontology building. She is a member of the Syrian Computer Society.