Natural Language Processing (NLP) for Requirements Engineering (RE): A Systematic Mapping Study

Liping Zhao[†]

Department of Computer Science, The University of Manchester, Manchester, United Kingdom, liping.zhao@manchester.ac.uk

Waad Alhoshan

Department of Computer Science, The University of Manchester, Manchester, United Kingdom, waad.alhoshan@postgrad.manchester.ac.uk; Al-Muhammad inn Saud Islamic University, Department of Computer Science, Saudi Arabia, wmaboud@imamu.edu.sa

Alessio Ferrari[†]

Consiglio Nazionale delle Ricerche, Istituto di Scienza e Tecnologie dell'Informazione "A. Faedo" (CNR-ISTI), Pisa, Italy, alessio.ferrari@isti.cnr.it

Keletso J. Letsholo

Faculty of Computer Information Science, Higher Colleges of Technology, Abu Dhabi, United Arab Emirates, kletsholo@hct.ac.ae

Muideen A. Ajagbe

Department of Computer Science, The University of Manchester, Manchester, United Kingdom, muideen.ajagbe@manchester.ac.uk

Erol-Valeriu Chioasca

Exgence Ltd., CORE, Manchester, United Kingdom, erol@exgence.com

Riza T. Batista-Navarro

Department of Computer Science, The University of Manchester, Manchester, United Kingdom, riza.batista@manchester.ac.uk

Abstract

NLP4RE (Natural Language Processing for Requirements Engineering) is an area of research and development that seeks to apply natural language processing (NLP) techniques, tools and resources to the requirements engineering (RE) process, to support human analysts to carry out various linguistic analysis tasks on textual requirements documents, such as detecting language issues, identifying key domain concepts and establishing requirements traceability links. This article reports on a mapping study that surveys the landscape of NLP4RE research to provide a holistic understanding of the field. Following the guidance of systematic review, the mapping study is directed by five research questions, cutting across five aspects of NLP4RE research, concerning the state of the literature, the state of empirical research, the research focus, the state of tool development, and the usage of NLP technologies. Our main results are as follows: (i) we identify a total of 404 primary studies relevant to NLP4RE, which were published over the past 36 years and from 170 different venues; (ii) most of these studies (67.08%) are solution proposals, assessed by a laboratory experiment or an example application, while only a small percentage (7%) are assessed in industrial settings; (iii) a large proportion of the studies (42.70%) focus on the requirements analysis phase, with quality defect detection as their central task and requirements specification as their commonly processed document type; (iv) 130 NLP4RE tools (i.e., RE specific NLP tools) are extracted from these studies, but only 17 of them (13.08%) are available for download; (v) 231 different NLP technologies are also identified, comprising 140 NLP techniques, 66 NLP tools and 25 NLP resources, but most of them – particularly those novel NLP techniques and specialized tools – are used infrequently; by contrast, commonly used NLP technologies are traditional analysis techniques (e.g., POS tagging and tokenization), general-purpose tools (e.g., Stanford CoreNLP and GATE) and generic language lexicons (WordNet and British National Corpus). The mapping study not only provides a collection of the literature in NLP4RE, but also, more importantly, establishes a structure to frame the existing literature through categorization, synthesis and conceptualization of the main theoretical concepts and relationships that encompass both RE and NLP aspects. Our work thus produces a *conceptual framework* of NLP4RE. The framework is used to identify research gaps and directions, highlight technology transfer needs, and encourage more synergies between the RE community, the NLP one, and the software and systems practitioners. Our results can be used as a starting point to frame future studies according to a well-defined terminology and can be expanded as new technologies and novel solutions emerge.

[†] Corresponding authors.

CCS Concepts:

• Software and its engineering→ Software creation and management→ Designing software→ Requirements analysis; • General and reference → Surveys and overviews

Additional Key Words and Phrases:

Requirements engineering (RE), software engineering (SE), natural language processing (NLP), systematic mapping study, systematic review

ACM Reference format:

Liping Zhao, Waad Alhoshan, Alessio Ferrari, Keletso J. Letsholo, Muideen Ajagbe, Erol-Valeriu Chioasca, Riza Batista-Navarro. 2021. Natural Language Processing (NLP) for Requirements Engineering (RE): A Systematic Mapping Study. ACM Comput. Surv. XXX, Article X. 33 pages.

1 Introduction

The important role of natural language (NL) in requirements engineering (RE) has long been established [1], [2]. In a survey published in 1981, aimed at providing an overview of techniques for expressing requirements and specifications, Abbott and Moorhead stated that "the best language for requirements is natural language [3]." While it is difficult to prove that NL is actually the best option, empirical evidence over the years has shown that it is at least the *most common* notation for expressing requirements in the industrial practice. The online survey of 151 software companies in the early 2000s by Mich et al. [4] concluded that in 95% percent of the cases requirements documents were expressed in some form of NL. This dominance of NL was confirmed by a recent survey of Kassab et al. [5], which involved 250 practitioners. The majority of the participants (61%) in that survey stated that NL was normally used in their companies for describing and specifying software and system requirements. Therefore, based on the past and current empirical evidence, we can safely assume that NL will continue to serve as the *lingua franca* for requirements in the future as well.

Inspired by the close relationship between NL and requirements, since the early 1980s researchers have been attempting to develop natural language processing (NLP) tools and methods for processing requirements texts [2]. However, for almost three decades, this research was hindered by inadequate NLP technologies, which did not work well enough at that time to support applications outside NLP research [6]. The situation has been changing dramatically since the late 2000s. Huge improvements and advances in NLP technologies, particularly the widely availability of NLP tools and resources [7], made it possible for researchers to explore a range of NLP enabled tools and methods for RE tasks [8], [9]. Research in NLP for RE, or *NLP4RE* for short, has since grown into an active, full-fledged research area [8], attracting researchers from the wider RE community. Today, research in NLP4RE is thriving and has a dedicated annual workshop, called NLP4RE [9]. There is now a great potential to develop effective NLP4RE tools that can serve the real-world practice of RE. This is particularly needed, as RE tasks are still manually performed in industrial practice, as witnessed by the data of the NaPiRE survey that shows that only 16% of the companies are using automated techniques for requirements analysis [10].

During this long history of NLP4RE research, many researchers have made landmark leaps in advancing our knowledge in this area and some of these unique contributions are briefly recorded here.

The first papers on NLP4RE research were published in 1983 by Chen [11] and Abbott [12], who proposed using syntactic features of English sentences for database modeling and program design. After these pioneering works, the beginning of 1990s saw some serious attempts to develop NLP4RE tools, among which were *findphrases* by Aguilera and Berry [13] and *OICSI* by Rolland and Proix [1]. For the remaining 1990s right up to the beginning of 2000s, a succession of NLP4RE tools had been proposed, among which were *AbstFinder* by Goldin and Berry [14], *NL-OOPS* by Mich [15], *Circe* by Ambriola and Gervasi [16], *CM-Builder* by Harmain and Gaizauskas [17], *QuARS* by Fabbrini et al. [18], and *ARM* by Wilson et al. [19]. Since the late 2000s, a large number of tools have been developed, among which are *SREE* (Tjong and Berry [20]) for ambiguity detection and *aToucan* (Yue et al. [21]) for model generation. More recent developments include tools for requirements classification [22], detection of defects [23], smells [24] and equivalent requirements [25], glossary extraction [26], requirements tracing [27], mining app reviews [28] and tweets [29], extracting user stories [30], and analyzing requirements-relevant legal texts [31].

However, in spite of this long history and increasing interest, except for a limited number of reviews that touch upon specific topics of NLP4RE (see Section 3), there has been no effort to provide a comprehensive view of the field as a whole. A holistic understanding of what has been done before, what is missing in the field, what are the strengths and weaknesses of existing studies, what overall trends are at play, and what these trends might mean, can steer the advancement of the field and facilitate the transition from research to industry. As Boote and Beile stated [32], a substantive, thorough and sophisticated literature review is a precondition for doing substantive, thorough and sophisticated research. To bridge this literature review gap in NLP4RE research, this article presents a large-scale systematic mapping study of the field. The mapping study reviews 404 relevant primary studies reported between 1983 and April 2019. We analyze the literature timeline and publication venues, the empirical maturity of the research, the RE phases addressed and tasks supported (e.g., tracing, defect detection, classification), the source documents used (e.g., requirements specifications, but also app reviews and legal text), the NLP4RE tools currently available and the most common NLP technologies adopted. Based on a structured classification scheme that covers these different aspects, we identify research trends and gaps, and we provide a terminological and conceptual framework to understand the field and guide future research. In particular, our results show that, although several solutions are proposed to address a large spectrum of RE phases, tasks and types of documents, there is still a relevant gap to cover in terms of industrial case studies and in terms of open-science practices, such as the sharing of datasets and tools. Furthermore, most of the NLP4RE solutions do not consider advanced NLP techniques for semantics and discourse analysis, indicating that NLP research has not yet been exploited at its full potential. Overall, our results show that it is now time to improve the empirical maturity of the field, with case studies, benchmarks and rigorous evaluations that can lead to more robust tools and open to industrial adoption of NLP4RE solutions.

The remainder of the paper is structured as follows. Section 2 sets the scene by introducing the concepts of NLP and NLP4RE. Section 3 presents the related reviews to show the lack of comprehensive understanding in the field. Section 4 describes the method for our mapping study while the mapping results are presented and analyzed in Section 5. Section 6 reflects on the mapping study and discusses its implications for future research and practice. Section 7 assesses the validity threats to this mapping study and our countermeasures. Finally, Section 8 concludes the paper.

2 Concepts and Definitions

2.1 Natural Language Processing (NLP)

NLP is a field that employs computational techniques for the purpose of learning, understanding and producing human language content [6]. Liddy [33] provides this definition:

Definition 1: Natural Language Processing is a theoretically motivated range of computational techniques for analyzing and representing naturally occurring texts at one or more levels of linguistic analysis for the purpose of achieving human-like language processing for a range of tasks or applications [*33*].

In this definition, the notion of "*levels of linguistic analysis"* refers to the *phonetic, morphological*, *lexical, syntactic, semantic, discourse,* and *pragmatic analysis* of language [33], the assumption of which is that humans normally utilize all of these levels to produce or comprehend language [34]. NLP systems may support different levels, or combinations of levels of linguistic analysis. The more levels of analysis NLP systems support, the stronger or more capable these systems are supposed to be.

The approaches to NLP can be broadly classified into *symbolic NLP* and *statistical NLP* [33]. Although both types of NLP have been investigated since the early days of the NLP field (circa 1950s), until the 1980s, it was the symbolic NLP that dominated the field.

Symbolic NLP emerged from artificial intelligence (AI). It is based on explicit representation of facts about language and associated algorithms and uses this knowledge to perform deep analysis of linguistic phenomena [33]. Symbolic NLP approaches include logic or rule-based systems, and semantic networks. In rule-based systems, linguistic knowledge is represented as facts or production rules, whereas in semantic networks, this knowledge is represented as a network of interconnected concepts.

However, symbolic NLP approaches lack the flexibility to dynamically adapt to new language phenomena, because they use the handwritten rules or the explicit representations built by human analysis of well-formulated examples to analyze input text [33]. Such rules may become too numerous to manage [35]. In addition, symbolic approaches may be frail when presented with unfamiliar, or ungrammatical input [33]. Beginning in the 1980s, but more widely in the 1990s, statistical NLP regained popularity, as a result of the availability of critical computational resources and machine learning (ML) methods [6]. Since then, statistical NLP has been the mainstream NLP research and development [36]. For example, many of today's NLP tools such as POS taggers and syntactical parsers are based on statistical NLP [37].

In contrast to symbolic NLP, statistical NLP employs various ML methods and large quantities of linguistic data (text corpora) to develop approximate, probabilistic models of language. These statistical models are simple and yet robust, because they are based on actual examples of linguistic phenomena provided by the text corpora, rather than deep analysis of linguistic phenomena as in symbolic NLP. When trained with large quantities of annotated language data, statistical NLP can produce good results, because it can learn most common cases in the copious data. Furthermore, the more abundant and representative the data, the better statistical NLP becomes.

However, statistical NLP can also degrade with unfamiliar or erroneous input [35], a problem similar to that of symbolic NLP. Furthermore, statistical NLP is mainly useful for low-level NLP tasks such as lexical acquisition, parsing, POS tagging, collocations, and grammar learning [33]. Today, many text and sentiment classifiers of statistical NLP are still solely based on the use of the words of a text to ascertain the meaning of the text, rather than using structure and semantics of the sentences or discourses of the text. Also, most statistical models are trained with text corpora of everyday usage language, such as WSJ (Wall Street Journal) articles. Consequently, statistical NLP can be unreliable for domain-specific text such as software and system requirements.

Most recently, circa 2012, deep learning (DL) methods began to emerge in the NLP scene [38]. The central idea of DL is that it allows a machine to be fed with a large amount of raw data and to automatically discover the representations or features needed for detection or classification [39]. Thus DL requires very little feature engineering by hand. Furthermore, the features learned by DL models are high-level, allowing for better generalization even over new, unseen data. By contrast, conventional ML techniques used in NLP are limited in their ability to process natural data in their raw form. This means that constructing a statistical NLP system requires careful engineering and considerable domain expertise to design a feature extractor that transforms the raw text into a suitable internal representation (i.e., feature vector), from which the ML subsystem, often a text classifier, can detect or classify patterns in the input. Both NLP and deep learning experts predict that NLP is an area in which deep learning could make a large impact over the next few years [6], [39]. Nonetheless, recent trends in deep learning based NLP show that coupling symbolic AI will be key for stepping forward in the path from NLP to natural language understanding [38]. This reaffirms the view that symbolic approaches and statistical approaches are complementary.

Our mapping study will focus on the application of NLP technologies (techniques, tools and resources) to NLP4RE, regardless whether they are based on symbolic or statistical NLP.

2.2 Natural Language Processing for Requirements Engineering (NLP4RE)

Based on the definition of NLP (Definition 1), we define NLP4RE as follows:

Definition 2: *Natural language processing for requirements engineering (NLP4RE) is an area of research and development that seeks to apply NLP technologies (techniques, tools and resources) to different types of requirements documents to support a range of linguistic analysis tasks performed at various RE phases.*

This definition has a number of key elements. First, we establish NLP4RE as *an area of research and development that seeks to apply NLP technologies to RE*. This has to be the precondition for NLP4RE, because NLP4RE is motivated and enabled by NLP. We differentiate between three types of NLP technology: NLP technique, NLP tool and NLP resource. A *NLP technique* is a practical method, approach, process, or procedure for performing a particular NLP task, such as POS tagging, parsing or tokenizing. A *NLP tool* is a software system or a software library that supports one or more NLP techniques, such as Stanford CoreNLP[1], NLTK[2] or OpenNLP[3]. A *NLP resource* is a linguistic data resource for supporting NLP techniques or tools, which can be a language *lexicon* (i.e., dictionary) or a *corpus* (i.e., a collection of texts). Existing lexicons include WordNet[4] and FrameNet[5], whereas examples of corpus include British National Corpus[6] and Brown Corpus[7].

Second, NLP4RE deals with *requirements documents*, as most of these documents are expected to be in NL. This is true for the later stages of RE, when requirements are documented and validated, but also in early phase RE, in which requirements analysts may have to consult a wide variety of documents in order to develop an understanding of the problem domain. Such documents include interview transcripts, legal documents, standards, and operational procedures [40]. More recently, online product reviews [28] have been found useful for understanding the needs and wants of end users. Consequently, the types of input to NLP4RE, referred here as requirements documents, are broad and diverse, and not limited to the classical requirements specifications only.

Third, while NLP strives for *human-like language processing,* to achieve human-like performance [33], NLP4RE has a less ambitious goal, as its main objective is to *support* requirements engineers to perform various RE tasks that involve processing and analyzing textual requirements documents [41], [42]. Such tasks include detecting language issues, identifying key domain concepts and establishing traceability links between requirements. As highlighted by Berry et al. [41], NLP tools for RE, or *NLP4RE tools* for short*,* can be more effective than humans in performing such clerical or data intensive activities. NLP4RE tools are therefore *RE specific NLP tools*, developed as a result of NLP4RE research and built on existing NLP technologies (tools, techniques and resources) to serve the purpose of RE activities.

We use this broad definition of NLP4RE to delineate the scope of our mapping study and to help identify relevant studies to NLP4RE.

3 Related Reviews

The RE literature counts several surveys with various degrees of relevance and quality, and covering some specific areas of interest. A survey of the progress made in empirical RE research since 1992 identified a total of 56 reviews, consisting of 7 mapping studies and 49 systematic reviews (Daneva et al. [43]), but none of them is related to NLP4RE. For example, the topics addressed by the reviews included in the survey are concerned with requirements writing standards, empirical evaluation of software requirements specifications techniques, service description methods, cloud computing security requirements, and success factors in enterprise resource planning systems. Given this knowledge gap in the

[1] https://nlp.stanford.edu/software/

[2] https://www.nltk.org

[3] https://opennlp.apache.org

[4] https://wordnet.princeton.edu/

[5] https://framenet.icsi.berkeley.edu/fndrupal/

[6] http://www.natcorp.ox.ac.uk/

[7] http://clu.uni.no/icame/manuals/

current RE literature, the raising interest in NLP4RE [8], [9], and the improvement of NLP technologies observed in the last years [7], motivated us to conduct our mapping study. In conducting the literature search for the study, we also identified 18 additional reviews that have some relevance to NLP4RE, but none of which was included in the 56 reviews selected by Daneva et al. [43]. The reviews are briefly discussed here to show the state of the secondary studies related to NLP4RE.

Among those 18 reviews, four of them focus on *modeling* activities in software engineering. In particular, Loniewski et al. [44] present a survey of RE techniques in the context of model-driven development, including the cases where model transformation involved the requirements expressed in NL. Yue et al. [45] provide a review of different techniques for transforming textual requirements into analysis models. In their review, NLP support for model transformation is also considered. On the other hand, the review presented by Nicolás and Toval [46] focuses on the techniques used to generate NL or formal requirements from models. The review by Dermeval et al. [47] covers the studies that use ontologies for requirements modeling and shows that most of the reviewed studies dealt with textual requirements by means of AI techniques, including NLP, to support different language analysis tasks.

Another group of reviews is concerned with topics related to *requirements management*, including retrieval, tracing and classification of requirements. Specifically, Irshad et al. [48] review papers on requirements reuse, including work that applies NLP and IR techniques to text similarity evaluation to match input queries with existing requirements in a repository. Torkar et al. [49] review the different methodologies to support traceability, considering also approaches that use some textual analysis support. On a different note, but still related to requirements management, Binkhonain and Zhao [50] review research on applying ML and NLP techniques to the classification of non-functional requirements and performance metrics used to assess different ML based approaches. More oriented towards software product line engineering, Bakar et al. [51] survey the usage of NLP techniques in the identification of common and variant features in NL requirements documents as well as other NL sources, such as NL product descriptions. Building on this survey, Li et al. [52] focus on identifying features and analyzing their relationships in textual requirements.

A rather recent, yet lively, group of reviews is concerned with analysis of publicly available *feedback* produced by users and developers. Among these reviews, Martin et al. [53] report a mapping study of the research on app store analysis in software engineering and identify NLP as relevant tools for feature analysis and app review analysis. Within the field of app review analysis, Tavakoli et al. [54] review the application of ML and NLP techniques to extracting and classifying useful information from users' feedback, so as to distinguish between requirements-relevant information and other types of users' comments. Two recent reviews by Santos et al. [55], [56] are concerned with classification of app reviews and users' feedback. Finally, with a broader focus on developers' feedback, Nazar et al. [57] review works on summarization of the various data sources, including bug reports, mailing lists and developer forums, to extract requirements related information.

Other reviews which might be relevant to NLP4RE, but more limited in scope and extension or topic-specific, are those by: Ahsan et al. [58] on test case generation from NL requirements and Garousi et al. [59] on NLP-assisted software testing; Shah et al. [60] on ambiguity detection in requirement also by means of NLP; Casamayor et al. [61], presenting a non-systematic survey on text mining and NLP in the field of model-driven design; and Nazir et al. [62], pursuing a mapping study on NLP for RE analogous to ours, but considering only 27 primary studies. Recently, also a MSc Thesis reported on a systematic literature review on using NLP for requirements elicitation and analysis [63]. Although very rigorous, the review is limited to both its scope, as it only focused on requirements elicitation and analysis, and the number of studies included, as it surveyed 144 studies.

This overview shows that while there are some literature reviews that touch upon specific topics of NLP4RE, they are limited in terms of coverage of the field. Indeed, each of the reviews investigates one specific NLP4RE-related topic, but there are no systematic synthesis and analysis of the surveyed topics and no extensive references that cover the broad area of NLP4RE. In addition, we note that our inclusion of these reviews in this overview is quite generous, intended to show the quantity of the existing reviews that have some degree of relation to NLP4RE, and thus the relevance of the field itself. Our mapping study aims to overcome these limitations by providing a comprehensive survey oriented towards a holistic understanding of the field. It not only provides a collection of the literature in NLP4RE, but, more importantly, also establishes a structure to frame the existing literature: through categorization, synthesis, analysis, and conceptualization of the main concepts and relationships that encompass both RE and NLP aspects. Our work therefore produces a *conceptual framework* of NLP4RE.

4 Review Method

To achieve our goal, we carried out a systematic mapping study [64] using the method presented in [65]. This section presents our research questions and describes the main activities involved in the mapping study; we report our mapping results in Section 5.

4.1 Research Questions

The research questions (RQs) for our mapping study are stated below. The RQs are interrelated, designed to interrogate the NLP4RE literature progressively. We first outline the main RQs in *italic*, and then we report in regular text specific questions that are used to elaborate each RQ. The answers to the set of specific questions provide the answer to the main RQ.

RQ1: What is the state of the literature on NLP4RE? Specifically, what is the population of the published literature on NLP4RE? What is the publication timeline? What are the leading publication venues?

RQ2: What is the state of empirical research in NLP4RE? Specifically, what primary types of research have been carried out in the area of NLP4RE? What primary types of evaluation method have been used in the research? What relationships can be observed between these research types and evaluation methods?

RQ3: What is the focus of the NLP4RE research? Specifically, what main RE phases have been addressed? What linguistic analysis tasks have been investigated for these phases? What is the relationship between these RE phases and tasks? What types of input document have been considered?

RQ4: What is the state of tool development in NLP4RE research? Specifically, what new tools have been developed? Which RE phases and tasks do these tools support? Which of these tools are available to the public?

RQ5: What kinds of NLP technologies have been used in NLP4RE research? Specifically, what NLP techniques, tools and resources have been employed? What RE tasks are targeted by NLP technologies?

4.2 Study Selection Process

4.2.1 Determining the Data Sources.

We identified the following digital libraries as the main data sources for our mapping study: ACM Digital Library (ACM); IEEE Xplore Digital Library (Xplore); ScienceDirect (SD); SpringerLink (SL). These libraries were chosen because they host the major journals and conference proceedings related to software engineering (SE) and RE. To complement these libraries, we also selected Association for Computational Linguistics Library (ACL), where the major contributions to NLP are likely to be published. These five libraries serve as our data sources for identifying the relevant literature. In addition, NLP4RE workshop was also identified as our search source because its proceedings were not published by any of the above digital libraries.

4.2.2 Formulating the Search Strategy.

Our search strategy was based on the direct search of the electronic databases of the aforementioned five digital libraries. The search terms for querying these libraries were constructed using the steps presented in [66]. Specifically, we used the major terms “requirements engineering” (representing the context of the research) and “natural language processing” (representing the intervention in this context) as the base terms; elaborate each base term with alternative spellings and synonyms; use the Boolean OR to incorporate synonyms, alternative spellings, alternative terms, and sub-field terms into each base term set, and Boolean AND to link the two sets of terms. Several iterations were performed to identify and refine the keywords. The complete set of the search terms is presented in Table 1.

TABLE 1: KEYWORDS FOR IDENTIFYING THE NLP4RE LITERATURE

Main Keywords	Derived Keywords
Requirements engineering	Requirements elicitation, requirements analysis, requirements specification, requirements modeling, requirements validation, requirements verification, requirements management, requirements traceability, requirements classification, requirements document, requirements specification
Natural language processing	NLP, statistical NLP, machine learning, deep learning, information extraction, information retrieval, text mining, text analysis, linguistic instruments, linguistic approaches

4.2.3 Performing the Literature Search.

Following a series of initial searches to fine-tune our search terms, we performed the main search in April 2019 on the aforementioned five digital libraries. For all the libraries except ACL, advanced search was conducted, whereas for ACL the search was conducted manually due to the lack of the advanced search feature. No publication timeline was imposed on these libraries and only the publications written in English were retrieved. The search results were imported into an Endnote library and combined with the initial search results. After automatically removing the duplicates, the total results were 11,489 (83 from the initial search and 11,406 from the main search). Finally, we conducted a targeted search on NLP4RE workshop proceedings (2018 and 2019) and Google Scholar, which identified 51 studies. Adding them to the search results in our Endnote library gave us a total of 11,540 items for study selection.

4.2.4 Selecting the Relevant Studies.

Based on the search results, we applied a set of inclusion and exclusion criteria (see Table 2) to systematically include relevant studies and exclude irrelevant ones. This study selection process, as shown in Figure 1, was performed in stages by a team of four *data inspectors* (Alhoshan, Letsholo, Ajagbe, Chioasca) and three *supervisors* (Zhao, Ferrari, Batista-Navarro). These stages are briefly described as follows.

TABLE 2: INCLUSION (I) AND EXCLUSION (E) CRITERIA FOR SELECTING RELEVANT STUDIES

I/E	No.	Criteria
I	1	Include peer-reviewed primary studies that are relevant to NLP4RE (cross-checking and validation needed for such studies).
I	2	If there are multiple relevant studies that report the same research, include the longest study only and exclude the rest of them.
E	1	Exclude tables of contents, editorials, white papers, commentaries, extended abstracts, communications, books, tutorials, non-peer reviewed papers, and duplicate papers.
E	2	Exclude short papers that have fewer than 6 pages if they are in a single column format.
E	3	Exclude reviews or secondary studies.
E	4	Exclude papers that are not relevant to NLP4RE based on title, abstract, keywords, introduction, and conclusion (cross-checking and validation needed for such papers).

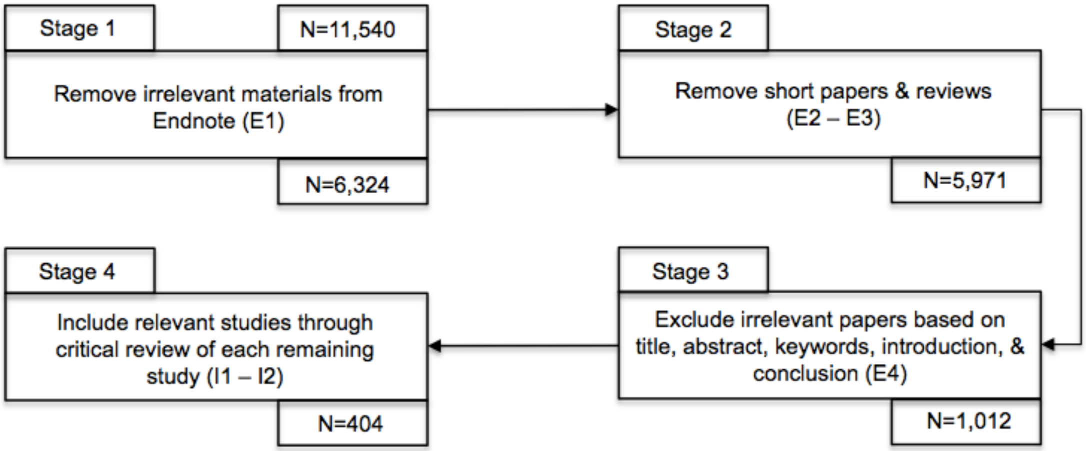
Stage 1
N=11,540
Remove irrelevant materials from Endnote (E1)
N=6,324
Stage 2
Remove short papers & reviews (E2 – E3)
N=5,971
Stage 3
Exclude irrelevant papers based on title, abstract, keywords, introduction, & conclusion (E4)
N=1,012
Stage 4
Include relevant studies through critical review of each remaining study (I1 – I2)
N=404

Figure 1: Study selection process.

In Stage 1, the lead inspector (Alhoshan) applied the exclusive criterion E1 (see Table 2) to the search results to filter out irrelevant contents such as editorials, commentaries and so on. This filtering stage left 6,324 studies in the library for further selection. In Stage 2, two inspectors simultaneously applied the exclusion criteria E2 and E3 to removed short studies[8] and secondary studies. Discrepancies or undecided cases from this stage were considered by the lead supervisor (Zhao) and the selection was then based on the majority voting agreement [67]. In Stage 3, all four inspectors autonomously performed study selection based on the exclusive criterion E4, with the support of the lead supervisor (Zhao) and using the majority voting for undecided cases.

In Stage 4, the remaining 1,012 studies were divided between the four data inspectors, who independently reviewed their allocated studies according to I1 and I2 to determine the relevance of each study. This involved carefully reading the full text of each study to establish its relevance and to identify its key components with respect to the predefined categories. During the selection, the three supervisors performed regular and random checking on the selected and discarded studies to ensure they were correctly included or excluded. Any discrepancies and inconsistencies were identified and immediately notified to the responsible inspectors. After individual selection, the inspectors crosschecked each other's results to decide about the relevance of each selected study. Where there was a disagreement, the lead supervisor was brought in to make the final decision. The selected studies by individual inspectors were then combined, resulting in a total of 416 studies. The lead supervisor carried out the final check on these 416 studies, by examining the title and abstract of each study, and in undecided cases, by reading the text of the study and consulting the other two supervisors. This identified 4 duplicate studies and 8 irrelevant studies. The final set to be included in our

[8] We have explicitly excluded the short study papers, because in general such studies lack detailed description of their contributions and including them can potentially skew the results of the mapping study.

mapping study thus comprised the remaining 404 studies. These 404 selected studies, together with our mapping study protocol, search methods and raw search results, are available online in our GitHub repository[9].

4.3 Data Extraction and Classification

4.3.1 Classification Scheme

Building a classification scheme is the central task of any mapping study, as the main purpose of this research method is to classify the literature [65], [68]. Our classification scheme reflects the five research questions. It is made up of four facets, each containing a set of categories. Figure 2 depicts this classification scheme, where the number associated with each category is the number of occurrences in that category, to be discussed in Section 5. The four facets and their categories are described as follows.

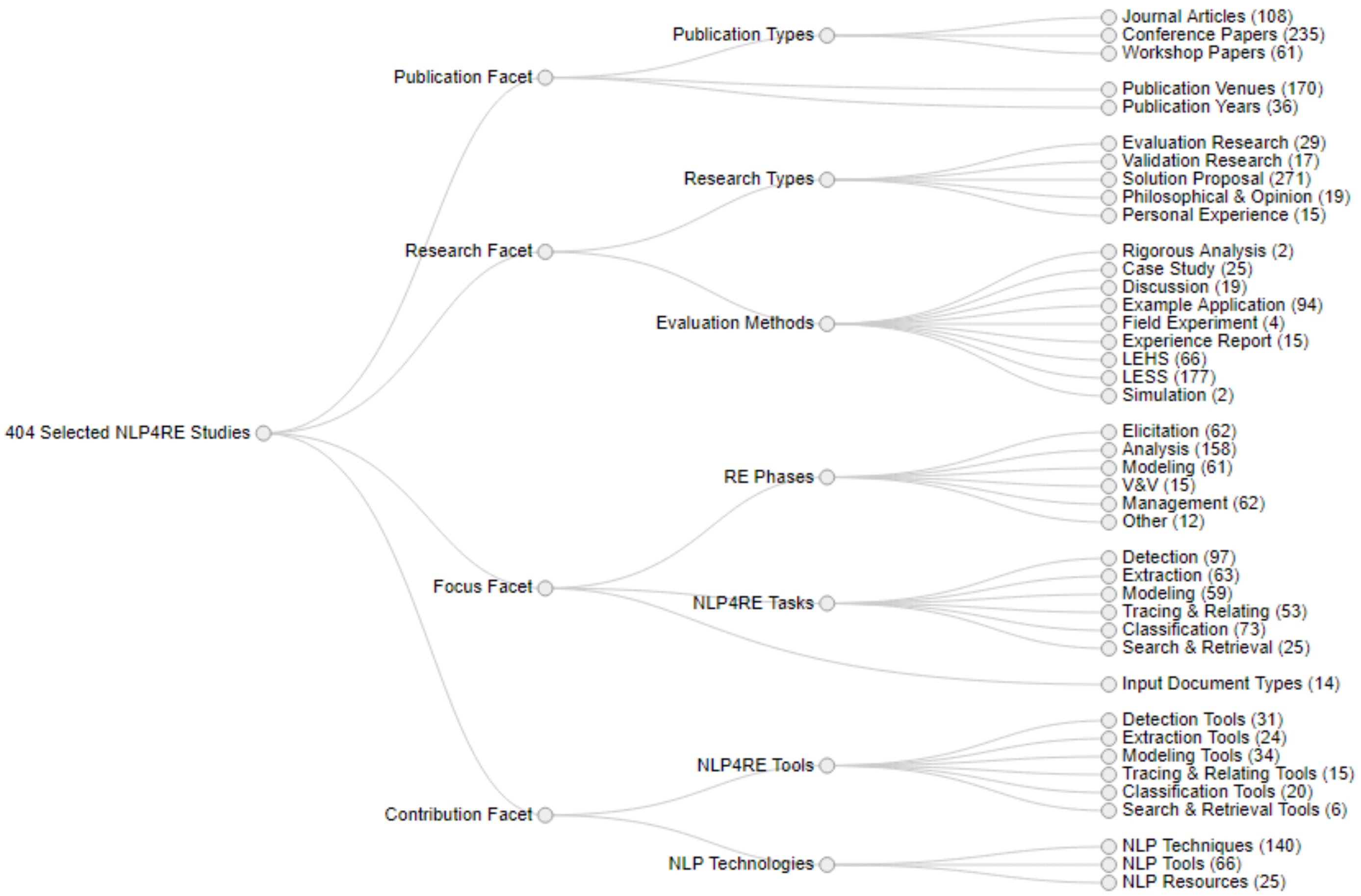

Figure 2: Faceted classification scheme for mapping the NLP4RE literature.

1. *Publication Facet.* This facet is for classifying the publication information. The resulting classification will be used to answer RQ1. This facet contains the following three categories: *Publication Types, Publication Venues* and *Publication Years*.
2. *Research Facet.* This facet is for classifying the selected studies according to their primary research types and primary evaluation methods. The resulting classification will be used to answer RQ2. This facet contains the following two categories:

- *Research Types.* Five types of research are considered, using the taxonomy of Wieringa et al. [69] (see Table 3).

[9] https://github.com/waadalhoshan/NLP4RE.

- *Evaluation Methods.* Nine types of evaluation methods are considered, according to Chen and Babar [70] (see Table 4).

3. *Focus Facet.* This facet is for classifying the selected studies according to their primary RE phases, NLP4RE tasks and input documents. The resulting classification will be used to answer RQ3. This facet contains the following three categories:

- *RE Phases.* We classify the NLP4RE studies according to six primary RE phases, as defined in Table 5. The first five phases are based on the work of Cheng and Atlee [71], whereas the last phase, "Other", is a placeholder, to be replaced by Testing or Design, or other phases of the software development process.
- *NLP4RE Tasks.* We differentiate six general NLP4RE task types in our mapping study, as shown in Table 6. These tasks involve applying linguistic analysis to RE documents. The first four of these tasks are roughly based on the work of Berry et al. [41], whereas the last two are derived from our observation of existing work. Specifically, Tracing & Relating task is performed to establish traceability links or relationships between requirements, and between requirements and other software artifacts such as models, code, test cases, and regulations (e.g., [72], [73], [74], [75], [27], [76]). Search & Retrieval is used to find requirements or requirements sets from existing repositories or documents (e.g., [77], [78], [79]).
- *Input Document Types.* The selected studies are classified into categories according to the input document type they use. The document types used in this classification are not predefined; they will be synthesized from the extracted data.

4. *Contribution Facet.* This facet is for classifying the contribution and the underlying technologies of the selected studies. The resulting classification will be used to answer RQ4 and RQ5. This facet contains the following two categories:

- *NLP4RE Tools.* These are the tools reported by the NLP4RE literature.
- *NLP Technologies.* NLP technologies used by the selected studies are classified into three types: NLP Technique, NLP Tool and NLP Resource.

TABLE 3: TYPES OF RESEARCH FOR CLASSIFYING THE NLP4RE LITERATURE (ADAPTED FROM WIERINGA ET AL. [69])

Research Type	**Explanation**
Evaluation Research	This type of research involves empirical evaluation from industries, often in the form of case study or field study.
Validation Research	This type of research, better called "assessment research", does not involve new technologies; instead, it is concerned with assessing existing technologies (methods, tools, algorithms etc.). Comparative studies are typically validation research.
Solution Proposal	This type of research typically involves the development of a novel solution (e.g., a new method, technique or tool) to a problem. This type of research may or may not contain validation research.
Philosophical/ Opinion	This type of research covers a broad range of work, under the guise of vision statements, position papers, opinions, viewpoints etc. This type of research is conceptual or theoretical, so its outcome is often a new understanding or a new perspective of some research area.
Personal Experience	This type of research is reflectional, based on personal experience of applying an existing technology to a real-world problem.

TABLE 4: TYPES OF EVALUATION METHOD FOR CLASSIFYING THE NLP4RE LITERATURE (BASED ON CHEN AND BABAR [70])

Evaluation Method	**Explanation**
Rigorous Analysis	Rigorous derivation and proof, suited for formal model.
Case Study	An empirical inquiry that investigates a contemporary phenomenon within its real-life context; when the boundaries between phenomenon and context are not clearly evident; and in which multiple sources of evidence are used.
Discussion	Provided some qualitative, textual, opinion.
Example	Authors describing an application and provide an example to assist in the description, but the example is "used to validate" or "evaluate" as far as the authors suggest.
Experience Report	The result has been used on real examples, but not in the form of case studies or controlled experiments, the evidence of its use is collected informally or formally.
Field Experiment	Controlled experiment performed in industry settings.
Laboratory Experiment with Human Subjects (LEHS)	Identification of precise relationships between variables in a designed controlled environment using human subjects and quantitative techniques.

Laboratory Experiment with Software Subjects (LESS)	A laboratory experiment to compare the performance of newly proposed system with other existing systems.
Simulation	Execution of a system with artificial data, using a model of the real word.

Table 5: RE Phases for Classifying the NLP4RE Literature (Based on Cheng and Atlee [71])

RE Phase	Explanation
Elicitation	This phase comprises activities that enable the understanding of the goals, objectives, and motives for building a proposed software system.
Analysis	This phase involves evaluating the quality of recorded requirements and identifying anomalies in requirements such as ambiguity, inconsistency and incompleteness.
Modeling	This phase involves building conceptual models of requirements that are amenable to interpretation.
Validation & Verification (V&V)	Requirements validation ensures that models and documentation accurately express the stakeholders' needs. Validation usually requires stakeholders to be directly involved in reviewing the requirements artifacts. Verification entails proving that the software specification meets these requirements. Such proofs often take the form of checking that a specification model satisfies some constraint (model checking).
Management	This is an umbrella activity that comprises a number of tasks related to the management of requirements, including the evolution of requirements over time and across product families, and the task of identifying and documenting traceability links among requirements artifacts and between requirements and downstream artifacts.
Other	This is an open-end category that allows us to record other NLP4RE related software development activities. For example, during software testing, NLP may be used to analyze requirements to generate test cases. In this case "Other" will be replaced by "Testing". During software design, NLP may be used to transform requirements into design artifacts. "Other" will be replaced by "Design".

Table 6: NLP4RE Tasks for Classifying the NLP4RE Literature (Adapted from Berry et al. [41])

NLP4RE Task	Meaning	Explanation
Detection	Detect linguistic issues in requirements documents	This task is typically to support manual review activities to make the requirements, or requirements-related artifacts, clear and unequivocal. The linguistic issues to be detected may range from the controversial usage of passive voice, to the occurrence of typically vague phrases (e.g., *as soon as possible*, *after some time*) or weak verbs (e.g., *may*, *could*), to the presence of syntactic and pragmatic ambiguities. Also checking the adherence to pre-defined requirements templates, and identifying equivalent requirements, can be included in this task, as the goal is still to enforce rigor in requirements texts.
Extraction	Identify key domain abstractions and concepts	This task normally aims to extract single or multi-word terms from requirements texts to establish domain-specific and project-specific *glossaries*, as requirements often include domain-specific, compound terms that are not commonly used. The extracted glossaries can be exploited for other objectives, including completeness or consistency checking, product comparison, classification, and modeling.
Classification	Classify requirements into different categories	This task aims to classify requirements into different types, base on the purpose for which the task is applied. For example, requirements can be categorized based on their *functional* category, to ease requirements apportionment and reuse or based on its *quality* category, to identify non-functional requirements that may be hidden within functional ones. Also, when applied to users' feedback and online discussions, classification can help identifying feedback that is specifically concerned with new requirements, or feedback referring to specific features of interest, possibly with the sentiment expressed by the product's users.
Modeling	Identify modeling concepts and constructing conceptual models	The task typically makes use of the extraction task, and can take different flavors, from the generation of UML models to support analysis and design, to the synthesis of feature models in a product-line engineering context, to the generation of high-level models of early requirements or user stories to support project scoping.

Tracing & Relating	Establish traceability links or relationships between requirements, or between requirements and other software artifacts	This task mainly aims to support manual tracing activities oriented to enforce and demonstrate process consistency, especially in a regulated context or in large-scale enterprise software. We include in this class also those works dealing with change impact analysis, as they also address the problem of identifying relationships between requirements or other artifacts.
Search & Retrieval	Search & retrieve requirements from existing repositories	The goal of this task can be to reuse existing requirements assets to match with the needs of novel customers, or to support domain scoping towards the development of new product, by recommending specific features based on existing software descriptions available online.

4.3.2 Data Extraction and Aggregation.

Our data extraction process was organized into four separate phases, described as follows. In Phase 1, we extracted the data for the categories and subcategories of the publication facet. These data were automatically obtained from our Endnote library where the selected studies were stored.

Phase 2 involved extracting the data for the categories and subcategories of the remaining three facets. To do so we first created a data extraction form on Google Sheets. The categories of the three facets (research, focus and contribution) were mapped onto the columns in the form. Some categories, i.e., RE phases and NLP4RE tasks, were given extra columns, to allow additional RE phases or tasks to be recorded. The rows of the form were used to record the data related to the selected studies, one row per study, and each row was identified by Study ID of a study. Next, we divided the selected studies evenly among the four data inspectors, who reviewed each study and performed data extraction for each study.

In parallel with the inspectors, the supervisors performed data extraction on the randomly selected studies and then compared the results with those obtained by the inspectors. Any discrepancy and inconsistency were discussed with the inspectors and actions taken to ensure the accuracy and consistency of the extracted data across the board.

After the completion of data extraction, in Phase 3, we carried out thematic synthesis on the descriptions related to input document types to identify specific input document types. We also performed coding [80], [81] and thematic synthesis [82] on the descriptions related to NLP techniques, NLP tools and NLP resources to create a coherent set of names for these technologies. This involves three steps [82]: code each piece of text with a label or term; translate related codes into the themes; organize the themes into high-level themes.

Finally, in Phase 4, we carried out data cleaning for each category to harmonize and standardize the terms in it; we inspected the members in each category to ensure our classification was accurate and consistent; we conducted statistical analysis of the categories; we employed a variety of visual tools [83], including tables, bar charts and dendrograms, to produce the visual representations of the analysis results [83], [84]. In the next section, we answer our five RQs based on the mapping results.

5 Mapping Results

5.1 RQ1: State of NLP4RE Literature

Population of the NLP4RE literature. A total of 404 primary studies relevant to NLP4RE are identified by our mapping study, comprising 26.75% (108) journal articles, 58.17% (235) conference papers and 15.10% (61) workshop papers. Such a trend, as we observed, is consistent with other areas in RE [85] and SE [86].

Publication timeline. These studies were published over the past 36 years, from 1983 to 2019 as Figure 3 shows (NB. The number of studies published in 2019 is incomplete, as our library search ended in April 2019); however, the majority of these studies (88.61% or 358 out of 404) have actually been published since 2004. This means that before 2004, an average publication rate was roughly two studies per year, whereas after 2004 it was about 24 studies per year. This rapid growth can be attributed to technological advances in NLP over the past 15 years or so [6].

Publication venues. These studies were found from 170 different publication venues[10]. The large number of diverse publication venues for these NLP4RE studies shows that NLP4RE has a core base in RE and a strong audience in SE; it has also attracted a general interest from diverse communities. Although this extensive number of diverse venues indicates that NLP4RE is a topic of general interest – though not necessarily central – to diverse communities, such a wide spread of the venues also has its downside, as it makes locating relevant NLP4RE work and building a consolidated NLP4RE knowledge base a difficult task.

Although the ACL library was included in our library search, we only found two relevant studies from it. The reason might be that the main publication venues for NLP4RE still focus on RE and SE, and that there is a lack of awareness of NLP4RE work in the NLP research community.

[10] The 404 selected studies and their publication venues are available as online-only material in the ACM Digital Library as well as in our GitHub repository (https://github.com/waadalhoshan/NLP4RE).

As NLP represents one crucial facet of NLP4RE, engagement with the NLP research community is important for NLP4RE researchers. One way of doing this is to take NLP4RE research results to NLP conferences or journals for discussion and feedback.

Of the 170 publication venues, 12 venues are identified as the leading venues for these studies (see Figure 4), as they collectively published 45.05% (or 182) of the selected studies. Thus, on average, each of these 12 venues published 15.17 studies, whereas each of the remaining 158 venues only published 1.41 studies. These 12 leading venues count some top RE and SE conferences and journals, as Figure 4 shows. It can be observed that the three top RE venues (RE, REFSQ and REJ) published a quarter of all selected studies, indicating that these are the core publication channels for NLP4RE research. As it can also be observed from Figure 4, the five top SE conferences and journals (JSS, ASE, ICSE, IST, and TSE) published more than a 10 percent of the studies, indicating that NLP4RE research has a broad audience in the SE community. To assess the impact of NLP4RE research in the SE community, we compare the number of NLP4RE studies published at ICSE versus the number of GORE (Goal-Oriented Requirements Engineering) studies published at ICSE, as GORE is a well-established subfield in RE with a long research tradition. Our mapping study found nine NLP4RE studies from ICSE, whereas five GORE studies from ICSE. Proportionally, NLP4RE-related ICSE studies account for 2.25% of the 404 studies included in our mapping study, while GORE related ICSE studies account for 2.03% of the 246 studies included in a recent mapping study on GORE [85]. This indicates that NLP4RE has made a similar impact on SE as GORE.

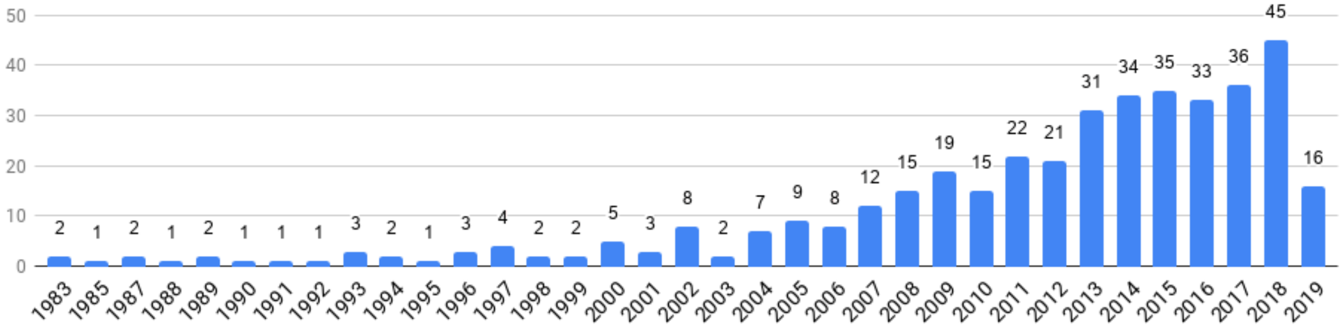

Figure 3: Publication timeline of the 404 selected studies.

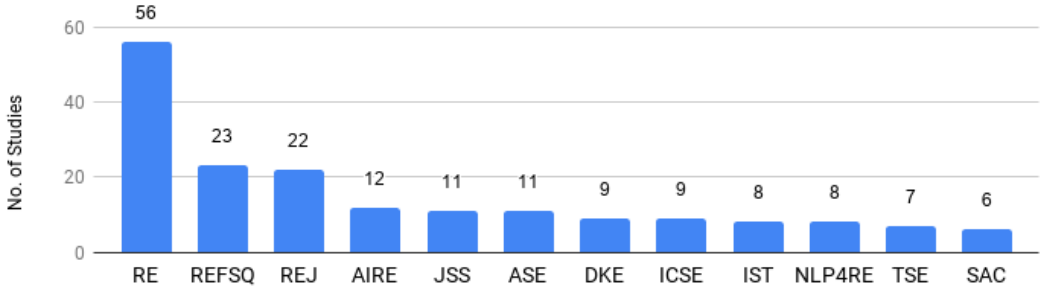

Figure 4: Leading publication venues for NLP4RE research.

These publication venues reveal that the majority of the NLP4RE studies are conference and workshop papers. Such a trend is consistent with the publication pattern in other RE and SE areas, and does not reflect a lesser quality across the field, as several RE and SE conferences such as RE and ICSE are highly competitive and comparable to top journals.

5.2 RQ2: State of Empirical Research in NLP4RE

To understand the state of the empirical research in NLP4RE, we classified the 404 studies according to their primary research types and their primary evaluation methods. We discuss the classification results as follows.

NLP4RE studies by primary research type. We classified the 404 studies exclusively according to their primary research types. What this means is that we associated each study with exactly one research type that we considered to be the primary research type of that study. This exclusive classification allows us to see the research trend in NLP4RE studies. The classification results (Figure 5) show that the majority of the NLP4RE studies (67.08% or 271) use Solution Proposal as the primary research type, whereas less than one fifth (17.33% or 70) use Validation

Research. The remaining three research types are less frequently used. We think this is a general trend, as it was also found in other RE and SE areas [85], [86].

NLP4RE studies by primary evaluation method. Similarly, to observe evaluation trends, we classified the 404 studies exclusively according to their primary evaluation methods, by associating each study with exactly one evaluation method that was considered to be the primary one of that study. The classification results (Figure 6) show that Laboratory Experiment with Software Subjects (LESS) is the most frequently used evaluation method, followed by Example Application and Laboratory Experiment with Human Subjects (LEHS). Other evaluation methods are far less common.

State of empirical research in NLP4RE. Empirical research is the one that seeks to explore, describe, predict, and explain natural, social, or cognitive phenomena by using evidence based on observation or experience [87]. Empirical research can be conducted in different ways, through, for example, systematic observation, experimentation, interview, survey, or examination of documents or artifacts [87], [88]. Among the nine evaluation methods used by NLP4RE studies, only five of them can be counted as empirical methods and these are LESS, LEHS, Case Study, Field Experiment, and Experience Report. The remaining four methods (i.e., Example Application, Discussion, Rigorous Analysis, and Simulation) are not empirical methods, because they are either conceptual (i.e., Example Application and Discussion), mathematical (i.e., Rigorous Analysis) or simulation exercises (i.e., Simulation) [89].

For the five empirical methods, Figure 6 shows that lab experiments (LESS and LEHS) are most common among the NLP4RE studies, whereas field studies (Case Study, Field Experiment, and Experience Report) are less common. This trend is consistent with empirical research in SE [90]. The main difference between a lab experiment and a field study lies in that the former is conducted in a controlled environment, typically with students (i.e., LEHS), whereas the latter is conducted in a real environment, typically with professionals [90]. Our results therefore show that most NLP4RE studies (more than 67%) are lab-based and only a small minority (7%) is conducted in an industrial setting. Although lab-based studies play an important role in gaining knowledge about new technologies [91], to advance knowledge, technologies must be evaluated rigorously in a real setting, by real users, because proving a tool works correctly (in the lab) does not equate to proving a tool works usefully for its intended audience and in its real-world context (in the field).

Evaluation methods coverage over research types. Using the pivot table (Table 7), we present the coverage of the primary evaluation methods over the primary research types in NLP4RE studies. Table 7 reveals some insights into the current state of NLP4RE research and the key takeaways from the table are: there is a clear alignment between different types of research and their evaluation approaches (this is good, but the alignment says nothing about the *quality* of the evaluation); a significant proportion of the solution proposals were assessed with a recognized empirical method (175/271 - good but not perfect); more than a third of the solution proposals were assessed only using an example application (94/271 – bad, call for action); only a small minority used human subjects (35/271 – bad, call for action); all 29 evaluation research studies were assessed in a real setting, using either a case study or a field experiment (29/29 – very good); among the evaluation research studies, case studies greatly outnumbered field experiments (25/29 – need further investigation).

Additionally, Table 7 also shows that *a typical NLP4RE study is a solution proposal, possibly evaluated internally through an experiment or example application, but without an external evaluation in the real world.*

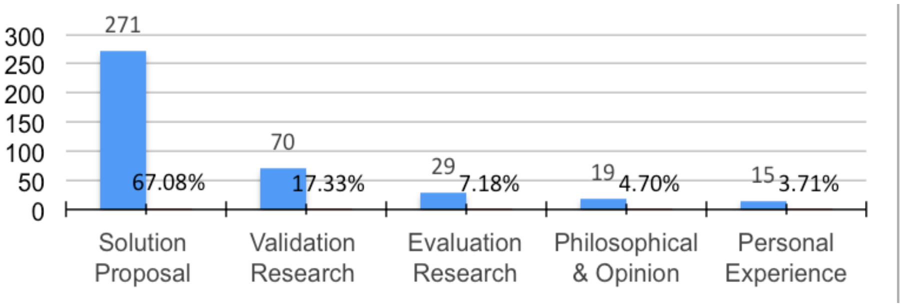

Figure 5: Distribution of the 404 studies according to their primary research types.

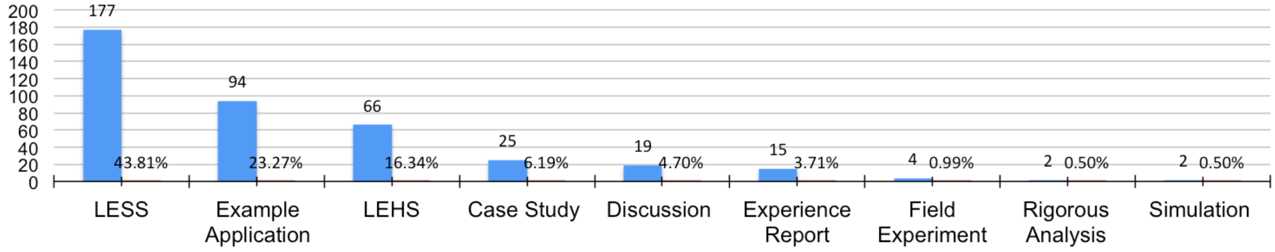

Figure 6: Distribution of the 404 studies according to their primary evaluation methods.

5.3 RQ3: Focus of NLP4RE Research

To address this research question, we exclude personal experience and philosophical & opinion papers, because they do not provide data pertinent to this question. We are therefore left with 370 studies for RQ3. These studies consist of 271 solution proposals, 70 validation research studies and 29 evaluation research studies.

TABLE 7: PRIMARY EVALUATION METHODS COVERAGE OVER PRIMARY RESEARCH TYPES IN THE 404 STUDIES

	Solution Proposal	Validation Research	Evaluation Research	Philosophical & Opinion	Personal Experience	**Row Total**
LESS	**140**	**37**	0	0	0	**177**
Example Application	**94**	0	0	0	0	**94**
LEHS	**35**	**31**	0	0	0	**66**
Discussion	0	0	0	**19**	0	**19**
Case Study	0	0	**25**	0	0	**25**
Experience Report	0	0	0	0	**15**	**15**
Field Experiment	0	0	**4**	0	0	**4**
Rigorous Analysis	0	**2**	0	0	0	**2**
Simulation	**2**	0	0	0	0	**2**
Column Total	**271**	**70**	**29**	**19**	**15**	**404**

Most targeted and least targeted RE phases. Figure 7 shows the distribution of the 370 studies by their main RE phases. Clearly, analysis is by far the most targeted RE phase, followed by management, elicitation and modeling. V&V (validation & verification), while testing and design are the least targeted phases. Analysis as the most targeted phase by NLP4RE research may be due to its broad role in RE. The lack of attention to V&V may be because it is a phase at the boundary between requirements and the rest of the software development process. The same can be said to testing and design phases. This shows that the main focus of NLP4RE are the phases within the main RE process, with limited attention to the relationship with the whole software process. This indicates an additional space for further research.

Note that about a third of the studies considered two different RE phases in their research, but the investigation of the second phase was often brief and secondary. In order to not skew the statistics, we focus only on the main RE phase targeted by each study.

Most and least studied NLP4RE tasks. Figure 8 shows the distribution of the 370 studies by their main NLP4RE tasks. Evidently, detection, classification, extraction, modeling, and tracing & relating are the most studied tasks, whereas search & retrieval is the least studied task. In what follows, we identify the focus of NLP4RE research through the relationships between these tasks and the primary RE phases where these tasks are performed.

Relationships between NLP4RE tasks and RE phases. For each RE phase, we count the number of studies that investigate each NLP4RE task. The results of the counting are presented in Table 8 from which some interesting observations can be made: from the RE phase perspective, analysis, management, elicitation, and modeling are targeted by a wide range of NLP4RE tasks. In addition, most of these phases has a central or *typical* NLP4RE task that represents the main role of NLP in that phase (detection for Analysis, extraction for Elicitation, tracing and relating for

Management, etc., see bold highlight in Table 8). Furthermore, (requirements) modeling and (software) design phases share exactly the same two NLP4RE tasks, with modeling as the central task for both phases. This suggests that, although the two phases occur at different stages of the development process, the main role of NLP in both modeling and design is essentially the same: to help identify requirements concepts and relationships. Likewise, from the NLP4RE task perspective, detection, classification, extraction, and tracing & relating are performed in a wide range of RE phases. In addition, classification, though not a central task for any RE phase (see bottom of Table 8), is more prevalent than tracing & relating, in spite of the latter being a central task for management. This perhaps due to the importance of classification and its general applicability throughout the whole RE process.

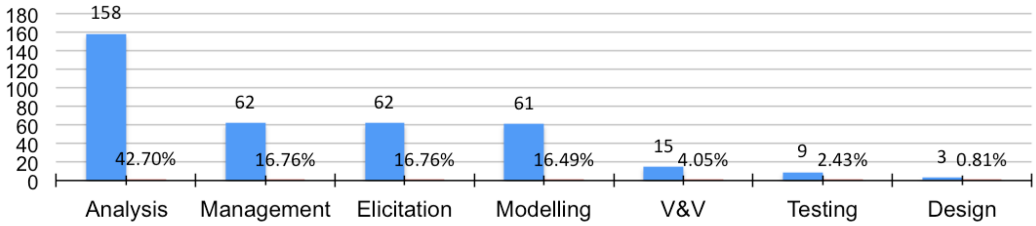

Figure 7: Distribution of the selected studies to different RE phases.

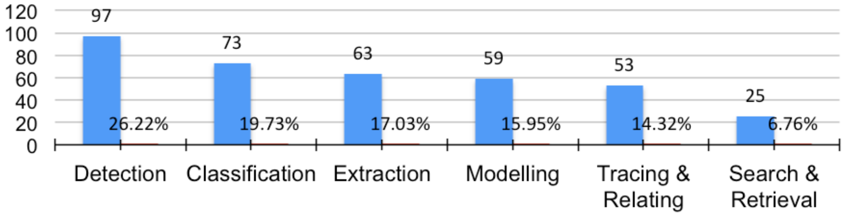

Figure 8: Distribution of the selected studies to different NLP4RE tasks.

Types of input document processed by NLP4RE studies. From the input documents extracted from our mapping study, we categorized them into 14 different types, as Figure 9 shows. For some input documents, we had to deduce and interpret their categories, due to their vague descriptions. Clearly from Figure 9, “requirements specification” is the most dominant input document type for NLP4RE research, used by 239 studies (64.59%), whereas “other model”, “code” and “UML diagram” are rarely used. Some document types are more recent, such as user-generated text, legal & policy document, user story, and domain document. For example, we found that processing user-generated text such as app reviews [28] and tweets [29] has become a fashionable trend in NLP4RE research since 2013 [92], but the trend appears to have peaked in 2017. Altogether, there were 26 NLP4RE studies (6.44% of the selected studies) that process this type of text. The reason for this trend could be that this kind of text is widely available, which can be used to test NLP4RE results, as a way to compensate the shortage of real requirements data [8]. However, processing this text may present new challenges to NLP4RE researchers: not only these documents do not conform to linguistic or structural standards that are normally expected for requirements specifications [40], but such documents could contain unfamiliar vocabulary and concepts to RE, more typical of conversational language. Given years of NLP4RE research, we believe that it is time for researchers to exploit more challenging texts and to explore uncharted territory.

Table 8: RE Phases and Corresponding NLP4RE Tasks

	Detection	Classification	Extraction	Modeling	Tracing & Relating	Search & Retrieval	**Row Total**
Analysis	**75**	37	16	0	19	11	**158**
Management	7	16	4	0	**26**	9	**62**
Elicitation	8	14	**32**	0	3	5	**62**
Modeling	0	0	3	**58**	0	0	**61**

V&V	5	4	2	0	4	0	**15**
Other (Testing)	2	2	4	0	1	0	**9**
Other (Design)	0	0	2	1	0	0	**3**
Column Total	**97**	**73**	**63**	**59**	**53**	**25**	**370**

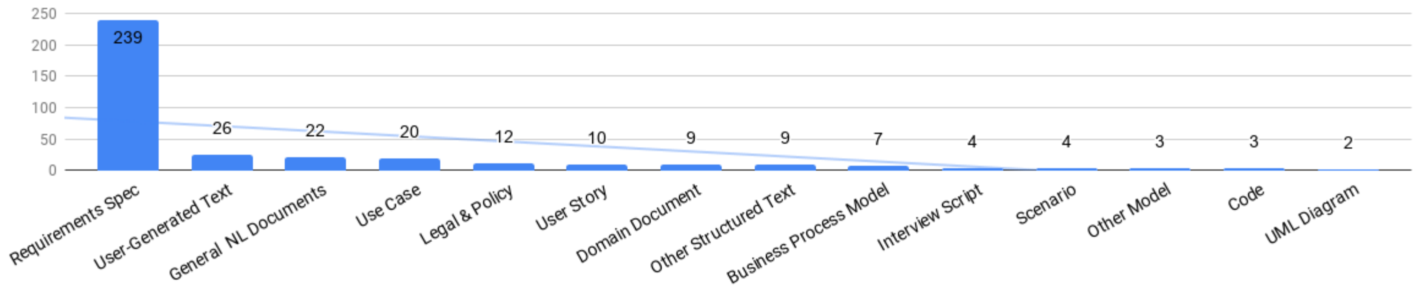

Figure 9: Input document types and the number of studies using each type.

5.4 RQ4: State of Tool Development in NLP4RE Research

From the reviewed studies that explicitly report new tools, we found 130 named tools. These tools were developed specifically to support the various NLP4RE tasks, so we call them *NLP4RE tools*, to differentiate them from *NLP tools*. In Table 9 these 130 identified NLP4RE tools are classified into six categories according to their main targeted NLP4RE tasks; each tool is represented by its name (e.g., OICSI) and indexed by a study identifier (e.g., S678). Through the study identifier of each tool, the reader can locate the reference of the study that reports the tool. Evidently, the top-ranked tools are modeling (34) and detection (31), followed by extraction (24), classification (20) and tracing & relating (15). At the bottom of the rank is search & retrieval (6).

By contrast, other types of tool, especially classification, detection and extraction, may be more readily composed from available NLP technologies and tools. For example, WEKA can be used to support classification, and GATE can be used to support extraction and detection. Consequently research on these tasks may focus on exploring different NLP techniques and tools, rather than on developing new tools. Finally, the limited number of search & retrieval tools can be attributed to the limited attention given to the search & retrieval task.

During our categorization of different tools, we noticed that some extraction tools also support other NLP4RE tasks such as paraphrasing and summarization. For example, extraction tools NAT2TEST and GuideGen perform both extraction and summarization tasks. The latter task is used to compose test cases or test guidelines.

Relationships between NLP4RE tools, NLP4RE tasks and RE phases. We use a circular dendrogram (Figure 10) to show the relationship between these tools and their corresponding RE phases. In this diagram, the NLP4RE tools are first grouped into the clusters by their NLP4RE tasks (the middle layer of the diagram) and then by their targeted RE phases (the inner layer of the diagram). This diagram can be used as a roadmap for us to navigate in both directions: from a given NLP4RE tool to its NLP4RE task and phase, and from a given RE phase to its tasks and available NLP4RE tools. Using this map, we can ask, for example, which NLP4RE tool is developed for which NLP4RE task and in which RE phase. Based on our results no tools were found for design phase. Furthermore, this map clearly illustrates the aforementioned *typicality* of NLP4RE tasks versus RE phases.

TABLE 9: CATEGORIES OF NLP4RE TOOLS

Tool Type	Tool Name (Study ID)	No. Tools	Percent
Modeling	OICSI (S678), NL-OOPS (S553), EA-Miner (S499), CM-Builder (S343), Circe (S34), LIDA (S623), NIBA Toolset (S272), RETNA (S108), aToucan (S909), DBDT (S31), Cico (S34), NL2UMLviaSBVR (S70), RADD-NLI (S121), SUGAR (S190), GRACE (S208), AREMCD (S219), RUCM (S227), RSLingo (S266), Zen-ReqConfig (S482), TREx (S496), NAPLES (S499), GeNLangUML (S551), ConstraintSoup (S600),	34	26.15%

	C&L (S707), AnModeler (S799), SBEAVER (S813), KCMP Dynamisch (S272), Xtext (S20), Kheops (S35), Visual Narrator (S683), ProcGap (S800), FeatureX (S772), CMT & FDE (S261), VoiceToModel (S765)		
Detection	ARM (S861), SREE (S812), RQA (S903), AnaCon (S41), REGICE (S55), NARCIA (S56), LELIE (S75), SRRDirector (S86), MIA (S114), KROSA (S178), NAI (S226), QuARS (S232), CAR (S252), CARL (S298), RAVEN (S303), ReqSAC (S370), RAT (S376), MaramaAIC (S395), RESI (S432), RECAA (S447), DeNom (S448), RETA (S450), AQUSA (S501), Dowser (S644), QAMiner (S661), LeCA (S701), S-HTC (S258), CNLP(S464), Pragmatic Ambiguity Detector (S256), ReqAligner (S663), REAssistant (S662)	31	23.85%
Extraction	findphrases (S13), AbstFinder (S307), FENL (S71), NAT2TESTSCR (S131), NLP-KAOS (S132), SAFE (S385), AUTOANNOTATOR (S433), UCTD (S453), GUEST (S598), Guidance Tool (S688), SpecQua (S743), NAT2TEST (S744), semMet (S777), Test2UseCase (S810), OCLgen (S845), Text2Policy (S872), GaiusT (S888), SNACC (S891), Doc2Spec (S897), ARSENAL (S915), MaTREx tool (S284), ELICA (S2), CHOReOS (S520), GuideGen (S907)	24	18.46%
Classification	ASUM (S129), RUBRIC (S223), WCC (S257), NFR2AC tool (S306), ALERTme (S332), PUMConf (337), FFRE (S341), AUR-BoW (S500), SEMIOS (S550), CRISTAL (S629), CoReq (S672), SD (S674), ACRE (S757), SOVA R-TC (S778), SMAA (S788), CSLabel (S892), HeRA (S718), NFR Locator (S758), SURF (S910), NFRFinder (S647)	20	15.38%
Tracing & Relating	Coparvo (S24), Trustrace (S25), Histrace (S25), CoChaIR (S26), HYPERDOCSY (S38), ReqSimile (S171), LGRTL (S198), CQV-UML (S400), TiQi (S651), REVERE (S717), LiMonE (S723), ESPRET (S792), COCAR (S805), RETRO (S934), WATson (S302)	15	11.54%
Search & Retrieval	RE-SWOT (S174), IntelliReq (S602), ReqWiki (S711), iMapper (S784), PriF (S802), WIKINA (S686)	6	4.62%
Total		**130**	**100%**

Tool development timeline. The 130 NLP4RE tools were reported between 1990 and April 2019 and their development timeline is shown in Figure 11. Clearly, before 2004, the development had been patchy, with just 18 tools produced; from 2004 onwards, however, there has been a year-on-year growth of NLP4RE tools, with only a brief dip in 2007 (the number of tools in 2019 is incomplete as our search was completed in April 2019). We observe that this growth period corresponds to the strong growth period of NLP4RE research (Figure 3), a clear indication of the maturity of NLP4RE research. We expect to see a large increase in tool development in the coming years.

Tool availability. Only 17 (13.08%) of these tools can be found online (See Table 10). However, of these 17 tools, aToucan requires access permission while IntelliReq is not accessible. For the remaining 15 tools, nine are open source tools hosted respectively on GitHub, SourceForge and a Semantic Wiki; three provide web-interface for users to try and another three can be downloaded from the Internet and installed on user computers. Among these accessible tools, only ReqSimile resisted the test of time, while the others were developed recently. Clearly, the state of the availability of NLP4RE tools is rather poor. We therefore hypothesize that the vast majority of NLP4RE tools were developed for the purpose of proof of concept, rather than for the purpose of practice.

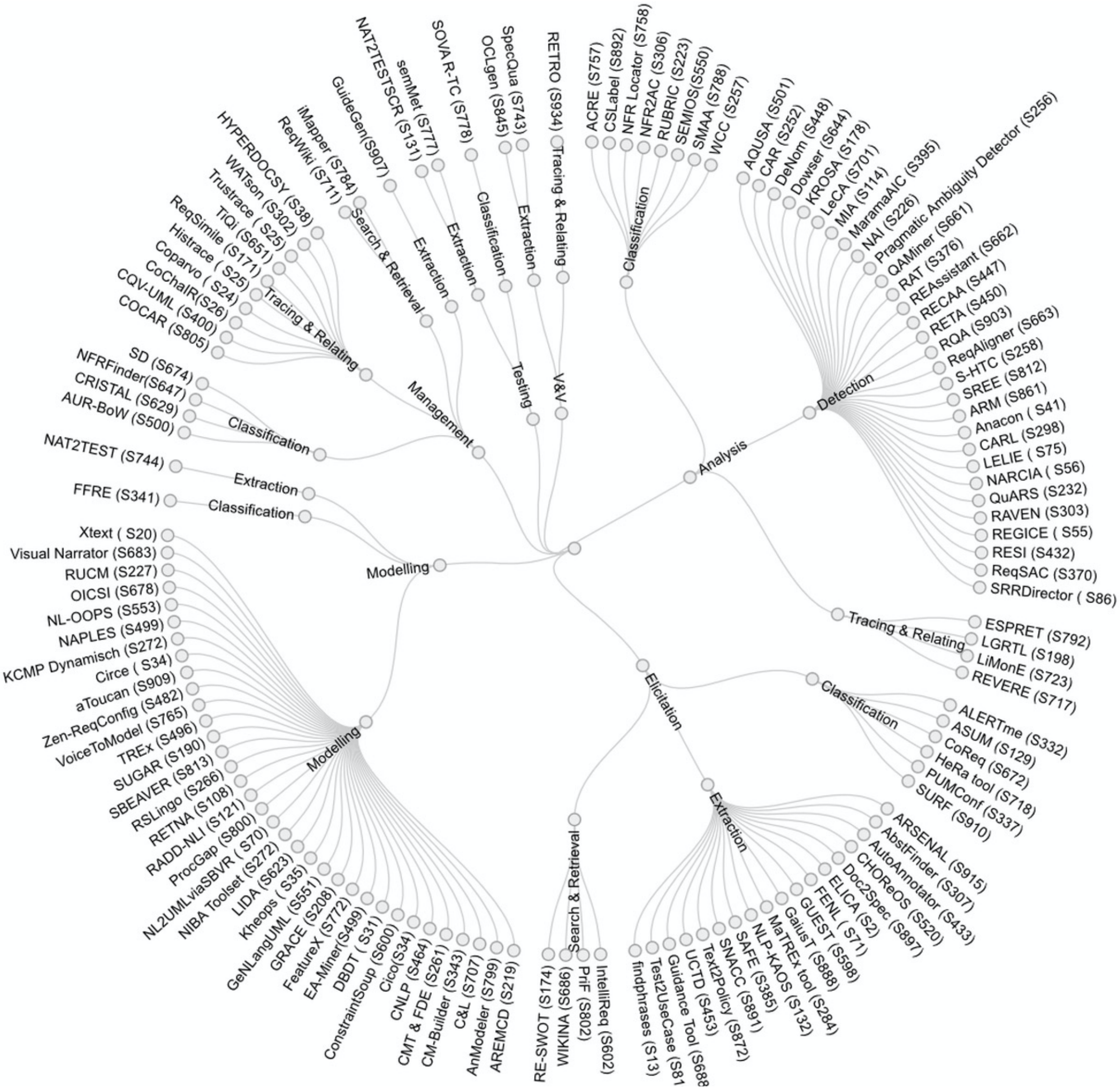

Figure 10: The 130 NLP4RE tools clustered by NLP4RE tasks and then by RE phases.

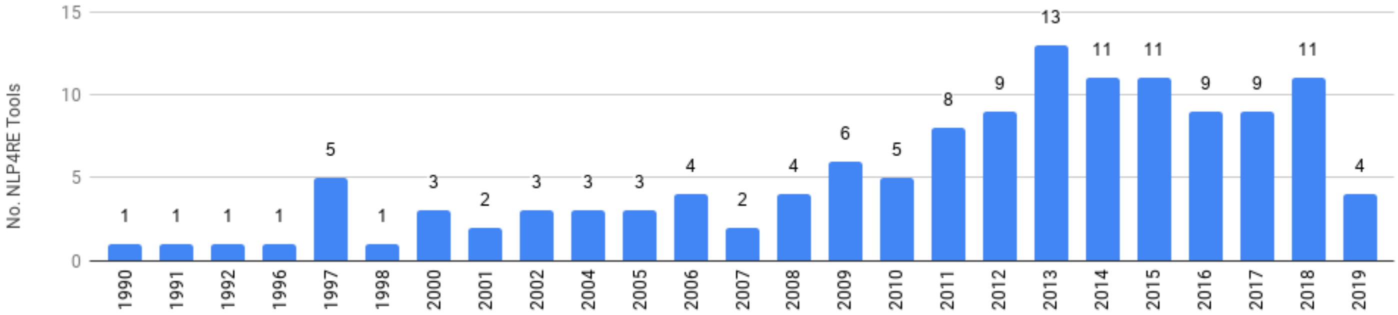

Figure 11: **Development timeline of NLP4RE tools.**

TABLE 10: NLP4RE TOOLS AVAILABLE ONLINE

Tool Name	Tool Type	Year	Web Address	Status
aToucan (S909)	Modeling	2015	https://sites.google.com/site/taoyue/atoucan-models	Need access permission
CMT & FDE (S261)	Modeling	2015	https://github.com/isti-fmt-nlp/tool-NLPtoFP	Free open source software
Visual Narrator (S683)	Modeling	2016	https://github.com/MarcelRobeer/VisualNarrator	Free open source software on Github
AnModeler (S799)	Modeling	2016	https://sites.google.com/site/anmodeler/	Software can be downloaded
FeatureX (S772)	Modeling	2018	https://github.com/5Quintessential/FeatureX	Free open source software
SpecQua(S743)	Extraction	2014	http://specqua.apphb.com	Free to try only, with a simple UI
Text2UseCase (S810)	Extraction	2019	https://sites.google.com/view/text2usecase/home	Web-based application, free to try, professional look and feel
GuideGen (S907)	Extraction	2019	https://github.com/hotomski/guidegen	Free open source software
Pragmatic Ambiguity Detector (S256)	Detection	2012	https://github.com/isti-fmt-nlp/Pragmatic-Ambiguity-Detector	Free open source software
IntelliReq (S602)	Detection	2014	http://www.intellireq.org	Website blocked
NARCIA (S56)	Detection	2015	https://sites.google.com/site/svvnarcia/	Can be installed on user computers
AQUSA (S501)	Detection	2016	https://github.com/RELabUU/aqusa-core	Free open source, with a command line UI
NFR Locator (S758)	Classification	2013	https://github.com/RealsearchGroup/NFRLocator	Free open source software
PUMConf (337)	Classification	2018	https://sites.google.com/site/pumconf/	Can be installed on user computers
ReqSimile (S676)	Tracing & Relating	2005	http://reqsimile.sourceforge.net	Free open source software, Beta version
ReqWiki (S711)	Search & Retrieval	2013	http://www.semanticsoftware.info/reqwiki	Free open source web-based application
RE-SWOT (S174)	Search & Retrieval	2019	https://github.com/RELabUU/re-swot	Free open source software on Github and visualisation on Tableau[11]

[11] The outputs of RE-SWOT can be viewed using Tableau tool: https://public.tableau.com/en-us/s/ (accessed 18 December 2020).

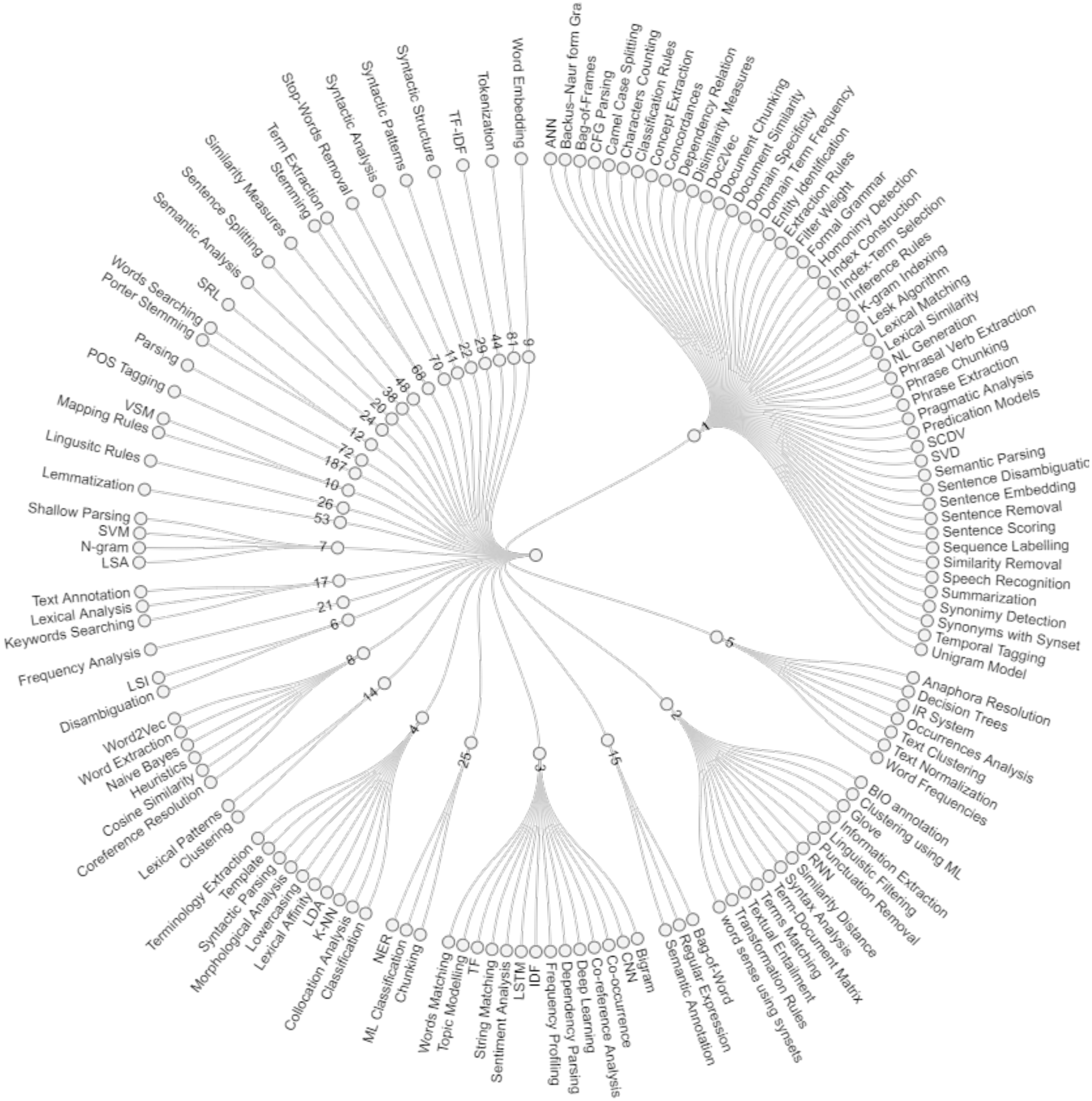

Figure 12: **140 NLP techniques and their frequency of use.**

5.5 RQ5: Use of NLP Technologies for NLP4RE

From the studies that make explicit reference to NLP technologies, we extracted and synthesized 231 different technologies, comprising 140 NLP techniques (Figure 12), 66 NLP tools[12] (Figure 13) and 25 NLP resources (Figure 14). These figures are briefly explained as follows.

Figure 12 shows that the most frequently used NLP technique is POS tagging (used by 187 studies), while the next most used techniques are tokenization (by 81 studies), parsing (by 72 studies), stop-words removal (by 70 studies), term extraction (by 68 studies), and stemming (by 68 studies). Figure 12 reveals that most NLP techniques, including those highly used, are syntactic techniques, a strong indicator of their dominance in NLP4RE research. Figure 12 also shows a large number of techniques are underused.

Figure 13 shows that the most frequently used NLP tool is Stanford CoreNLP (used by 80 studies), while the next most used tools are GATE (by 35 studies), NLTK (by 23 studies), Apache OpenNLP (by 21 studies), and WEKA (by 13 studies). Among these, apart from WEKA, which is a data-mining tool, the other four tools are general-purpose NLP tools. Figure 13 also reveals a large number of underused tools.

In Figure 14, the most frequently used NLP resource is WordNet (by 66 studies), followed by VerbNet (by 9 studies) and British National Corpus (by 7 studies). Most NLP resources listed in Figure 14 are lexical resources. There are a large number of underused resources.

Long tail distribution of NLP technologies. The usage of NLP technologies (techniques, tools and resources) as shown in Figure 13, Figure 14, Figure 15 exhibits a distribution pattern known as the "Long Tail" [93]. It means that only a small number of technologies were used frequently, as described above, whereas the majority technologies had a very low usage. The most frequently used technologies are called the "hits", whereas the least used are called "long tails". Clearly, POS tagging, Stanford CoreNLP and WordNet are the hits, indicating their popularity in NLP4RE research. However, what is more interesting is the huge number of long tail technologies – particularly those that are only used once or twice. According to the Long Tail theory [93], such technologies are "niches" in the market, but that is not entirely true of the NLP technologies used in NLP4RE research. Some of the long tail technologies are discussed in the following.

Long tail NLP techniques. Our initial investigation suggests that most long tail NLP techniques are *nascent*, so their application in NLP4RE might be forthcoming. For example, various deep learning techniques such as Word Embedding, Doc2Vec, LSTM, CNN, and RNN, are novel. Google's vector representation of words (Word2Vec) was only developed in 2013 [94]. It is therefore natural that they have a very low usage. Some long tail techniques are probably out of date or obsolete. For example, Lesk Algorithm, a classical algorithm for word sense disambiguation (WSD), is now being replaced by more advanced WSD techniques [95]. A few long tail technologies are real *niches*, as they are developed to serve specific needs. For example, CFG Parsing is a formal grammar for regulation expression and SCDV is a specialized technique to support faster construction of feature vectors in ML algorithms. For NLP4RE researchers, nascent techniques provide the springboard for innovation and novelty. In the future there will be more and more long tail NLP technologies, but only a small number of them will become future hits, according to the Long Tail theory [96]. A particularly promising one is the BERT language model [97], which is being used in recent RE research [98], [99].

Long tail NLP tools. Most long tail NLP tools are competing tools, such as taggers (e.g., Genia Tagger, CLAWS POS Tagger and Brill's POS Tagger) and parsers (e.g., ANTLR Lexical Parser, SEMAFOR Parser and Java SAX Parser). We argue that these tools are less used because they are less known by NLP4RE researchers and they are overshadowed by all in one toolkits such as Stanford CoreNLP and GATE.

Long tail NLP resources. According to our research, long tail NLP resources tend to be domain specific or language specific. For example, VerbOcean is a lexicon for mining semantic verb relations on the web; MODIS and CM-1 are datasets for software engineering [100]; GermaNet (a lexicon for German language) and Floresta Corpus (a syntactic tree corpus for Portuguese) are Language-specific resources. A few long tail resources are nascent, such as Google News Corpus and DBpedia corpus for Wikipedia. There are also a couple of out of date resources, such as Brown Corpus and Hornby's Verb Patterns. Please note that other RE specific resources may have been published in short papers and thus have not been included in our study. Important cases are the PURE dataset [101], the SEM-REQ dataset [102], the datasets for traceability discussed by [103], and the user story dataset [104] - used by a recent study [105].

NLP techniques and NLP4RE tasks. As it is not possible to show the relationship between all the 140 NLP techniques and their supporting NLP4RE tasks, we choose the 32 top-ranked NLP techniques for discussion. Figure 15 shows these techniques and their frequency of use – each technique is used at least 10 times. Figure 16 depicts these techniques and their relationships with the six NLP4RE tasks. It shows that these NLP techniques are evenly distributed across all six NLP4RE tasks, revealing a symmetric pattern. This suggests that all these 32 techniques are generally applicable to the six NLP4RE tasks. However, as these techniques are predominantly word-based, we deduce that most current NLP4RE studies are based on basic syntactic analysis, with little consideration to semantic or discourse analysis.

NLP tools and NLP4RE tasks. As with NLP techniques, we choose the top 14 NLP tools for discussion, on the basis that each tool is used at least three times. Figure 17 shows these tools and their frequency of use. The relationship between these tools and the six NLP4RE tasks is depicted in Figure 18, which shows that most tools are used for detection, classification and extraction, and only a few for modeling, tracing & relating and search & retrieval. The lack of NLP tools for modeling thus supports our early claim that more modeling tools are needed. We notice

[12] NLP tools are not NLP4RE tools; they are general-purpose tools that support the development of NLP4RE tools.

that the aforementioned general-purpose NLP tools – that is, Stanford CoreNLP, GATE, NLTK, and Apache OpenNLP – were used to support all six NLP4RE tasks.

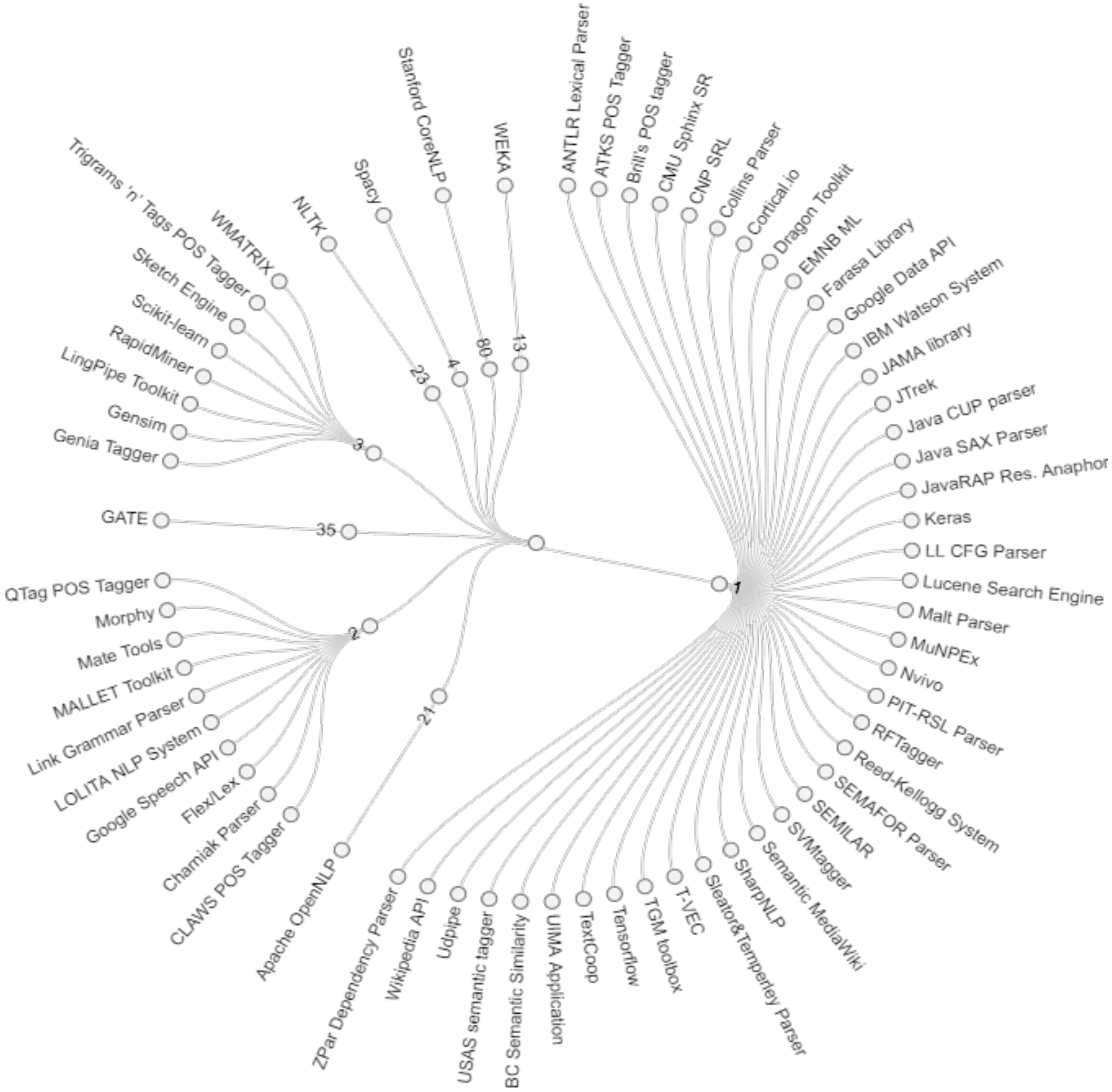

Figure 13: 66 NLP tools and their frequency of use.

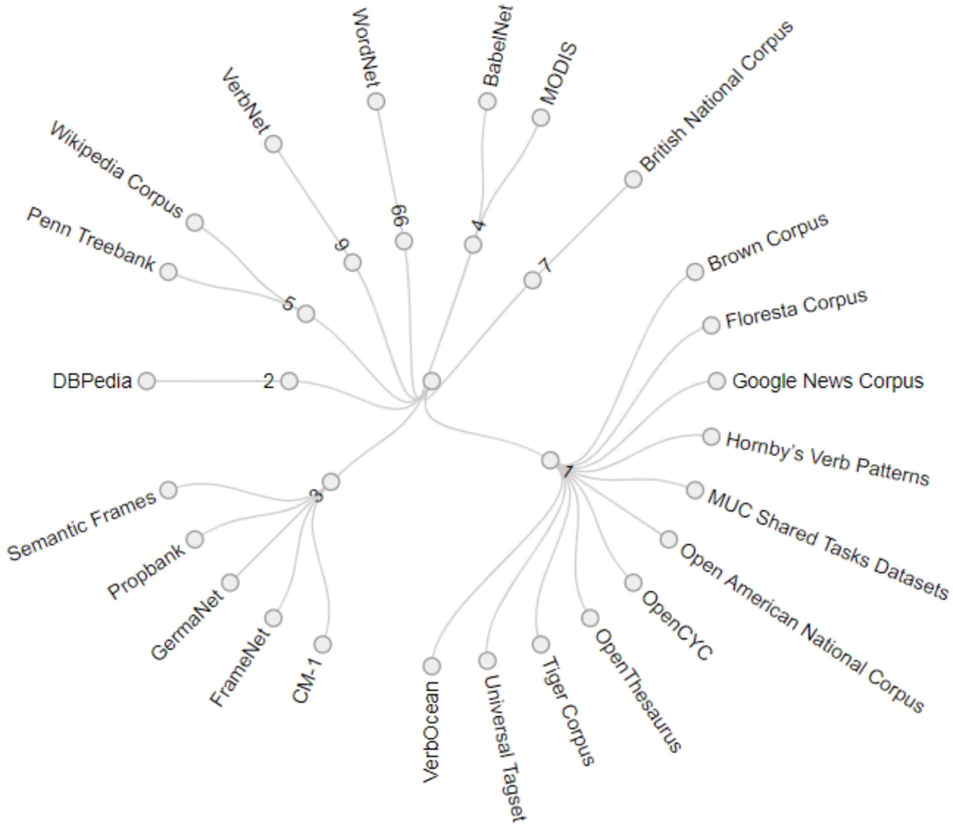

Figure 14: 25 NLP resources and their frequency of use.

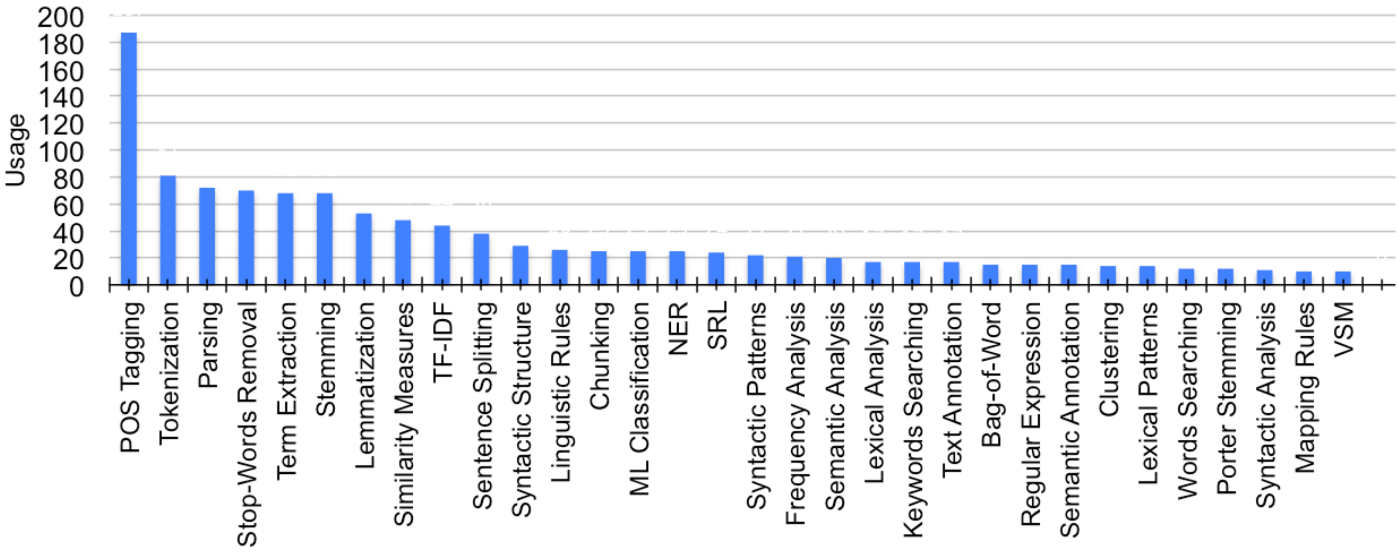

Figure 15: Frequently used NLP techniques - those that are used 10 times or more.

NLP resources and NLP4RE tasks. We choose 13 NLP resources for discussion, on the basis that each resource is used at least three times. Figure 19 shows these resources and their frequency of use. The relationship between these tools and the six NLP4RE tasks is depicted in Figure 20, which shows that resources are in general scarce for all the tasks, but particularly so for modeling and search & retrieval. Among these NLP resources, only WordNet is used to support all six NLP4RE tasks; VerbNet is used for all tasks but modeling; British National Corpus is used in all but modeling and search & retrieval. We observe that there is a general lack of NLP resources suitable for the NLP4RE tasks. This also suggests a general lack of *task specific* annotated RE dataset.

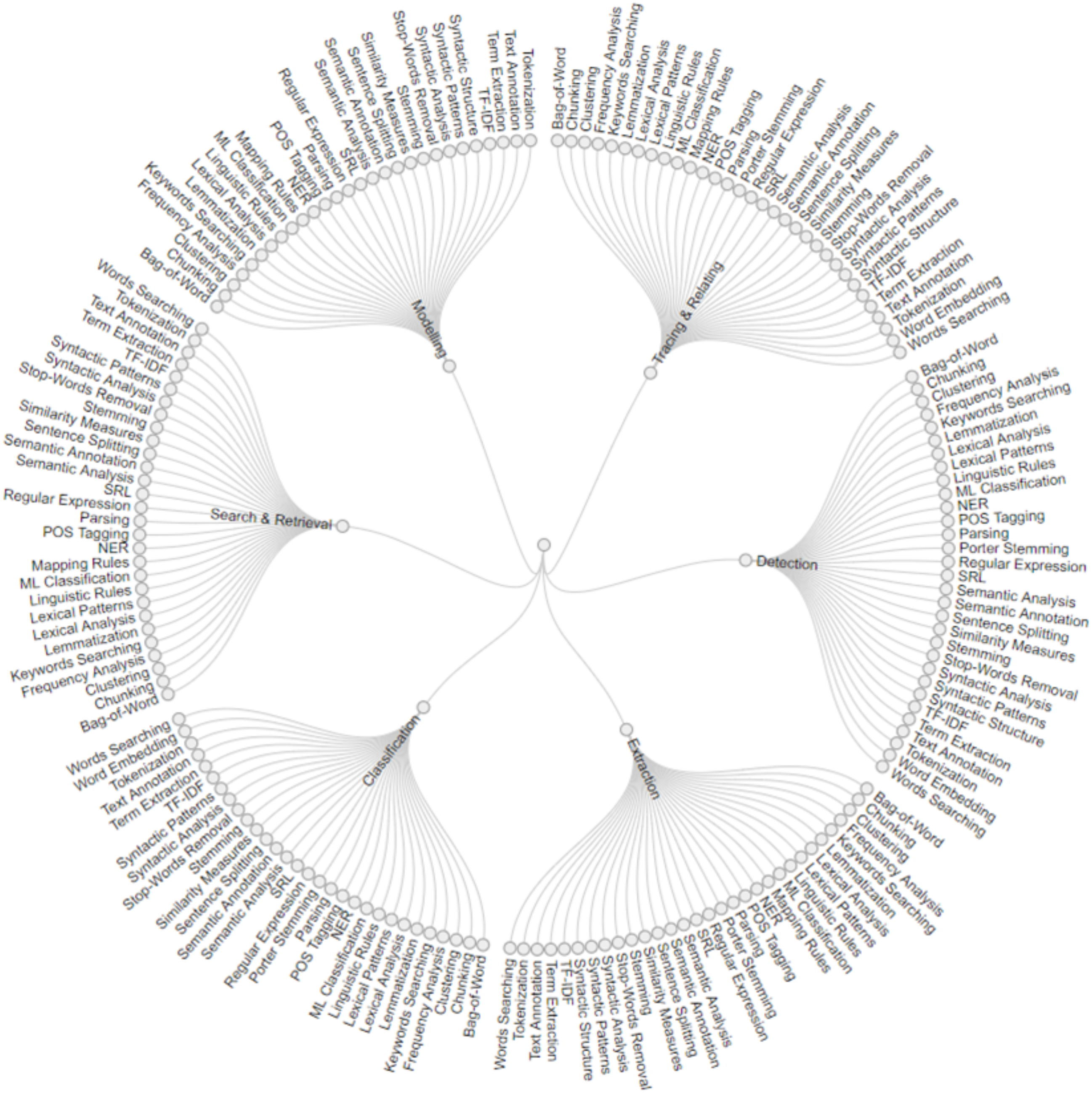

Figure 16: Relationship between frequently used NLP techniques and the 6 NLP4RE tasks.

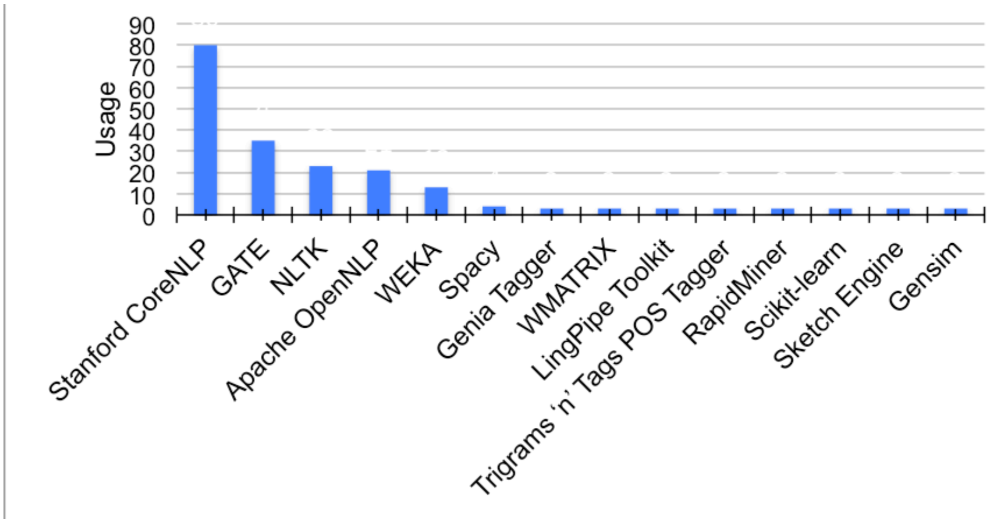

Figure 17: Frequently used NLP tools - those that are used three times or more.

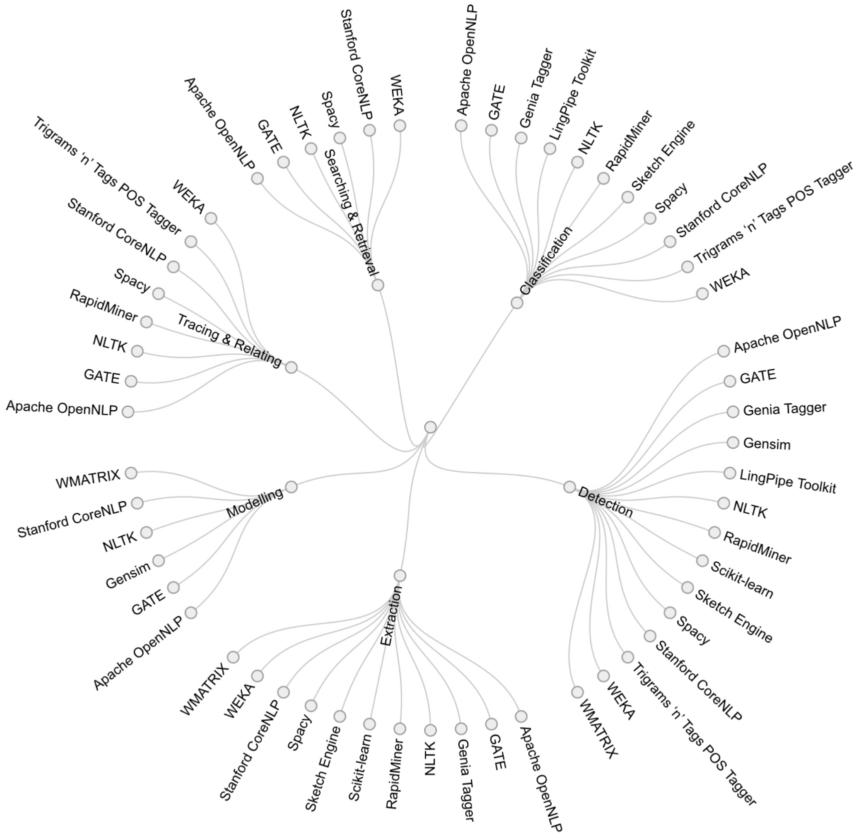

Figure 18: Relationship between frequently used NLP tools and the corresponding NLP4RE tasks.

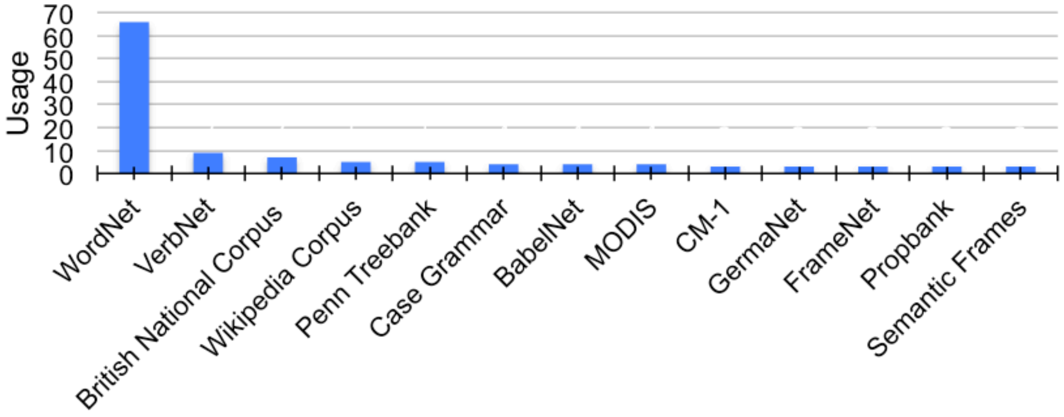

Figure 19: **Frequently used NLP resources - those that are used three times or more.**

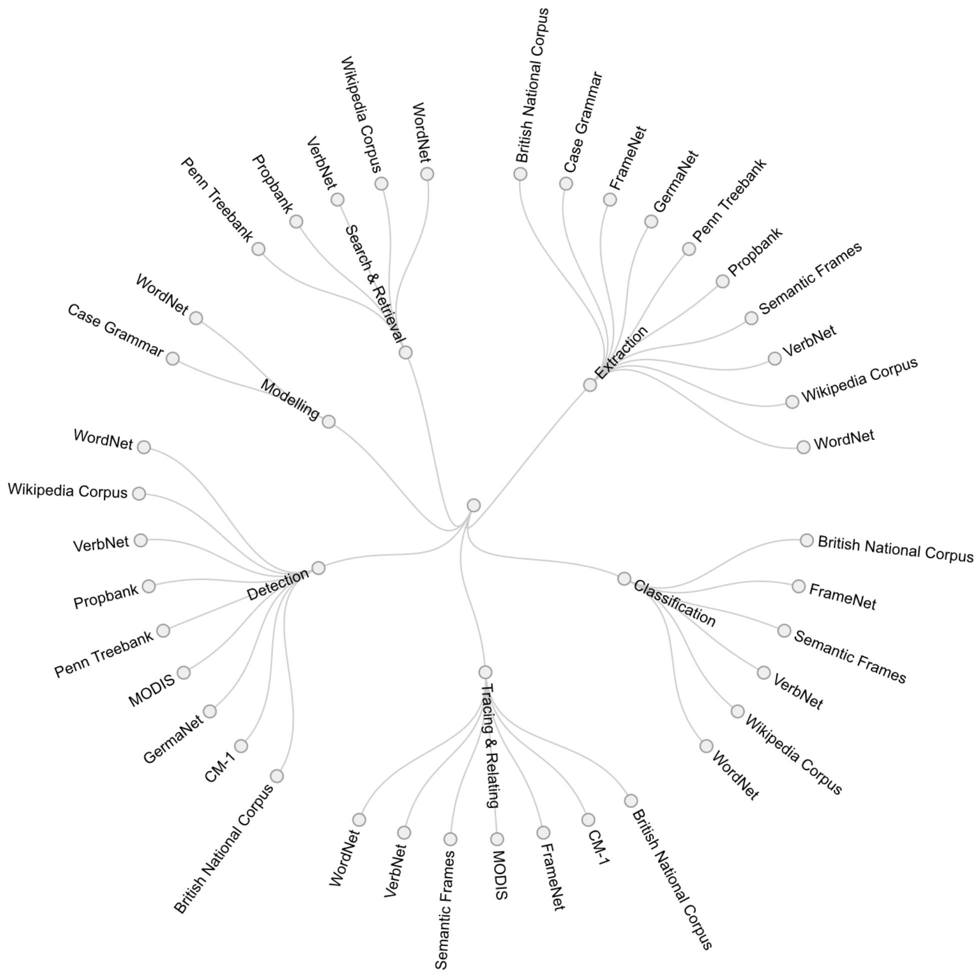

Figure 20: **Relationship between frequently used NLP resources and the corresponding NLP4RE tasks.**

6 Implications of the Mapping Study

Reflecting on the mapping study, here we discuss its implications for NLP4RE research and development and offer some suggestions for both NLP4RE researchers and practitioners.

The state of the NLP4RE literature (RQ1): Although NLP4RE studies can be found in a large number of publication venues covering diverse topics of computer science and information technology, the main venues for these studies remain firmly in the RE and SE areas (Sect. 5.1). This finding suggests that researchers looking for NLP4RE studies can safely focus on the RE and SE venues, where major works on NLP4RE are most likely published. However, there is one area outside both RE and SE that is inherently relevant to NLP4RE research, namely NLP. We believe that presenting NLP4RE work at relevant NLP conferences will enable us to mingle with NLP researchers for potential collaborations, and to better learn about most recent developments in NLP. In return, we also believe that NLP researchers can benefit from addressing the challenges of NLP4RE problems. We therefore suggest that the NLP4RE community should actively seek collaborations with the NLP community.

The state of empirical research in NLP4RE (RQ2): The mapping study shows (Sect. 5.2) that the majority (more than 67%) of NLP4RE studies are solution proposals, but more than one third of them are only evaluated by an example application whereas just one in eight of them are evaluated involving human subjects. This reveals that the status of evaluation in NLP4RE research is rather poor. We believe this is an area in which the NLP4RE community and the empirical software engineering community can collaborate. *We suggest that NLP4RE researchers should carefully choose a suitable evaluation methodology in their design of evaluation,* as there is a clear relationship between the quality of a study and the proper use of the evaluation methodology [70]. To this end, NLP4RE researchers need to understand and consult different evaluation methodologies. For lab experiments, we recommend to use the experimental design guidelines by Kitchenham et al. [91]; for case study and field experiment, we recommend the guidelines for conducting and reporting study research by Runeson et al. [106]; for user-centered evaluation, we recommend the design framework by Zhao et al. [88]. A comprehensive reference handbook for empirical software engineering is provided by Wholin et al [107]. Empirical evaluation in NLP4RE should also take into account the typical measures used in NLP (cf., e.g., Manning and Shutze [108]), although context-specific adaptations may be required (cf. *RQ5* below).

Our mapping study also shows that the majority (more than 60%) of NLP4RE studies are lab-based, and by contrast, only a small minority (7%) of them is conducted in an industrial setting. An obvious recommendation would be to call for the help from practitioners to evaluate NLP4RE research, but it is well known that it is usually difficult to get practitioners involved in research evaluation, due to time and other restrictions [90]. *While we believe that industrial case studies and field experiments play an important role in the success of the research, we suggest that when these are not possible, an alternative is laboratory experiments with human subjects (LEHS) and this often means the use of students as subjects*. Industrial experts accept that students are close to the software developer population [91], [90] and the use of students in the early stages of research evaluation is an effective way to advance software engineering theories and technologies [90].

The focus of NLP4RE research (RQ3): While most NLP4RE research focuses on the traditional RE phases (Sect. 5.3), especially the analysis phase, only a limited effort is dedicated to the phases that overlap with software development, such as design and testing. As the main purpose of RE is to support the transformation of user requirements into software systems, NLP4RE research can play an important role in this transformation, either by developing new design tools or new tracing tools. *NLP4RE researchers are therefore recommended to expand the research scope beyond the typical RE process*, possibly creating synergies with researchers working on NLP for feature extraction [109], [110] and on NLP for testing [111]. NLP4RE researchers sought to address NL issues related to a range of NLP4RE tasks (Sect. 5.3). While this mapping study identified six such tasks, it may have not uncovered other tasks, which can be investigated in future research. Additional areas of research come from the types of document that can be considered for evaluation: user stories, use cases, domain documents, interview scripts and models are still marginal in the research, while they have a primary role in practice. Furthermore, although recent research is giving relevant attention to artifacts such as user feedback, legal documents and user stories, the field is still open for further investigation.

The state of tool development (RQ4): In spite of the large number of tools produced by NLP4RE research, evidence of industrial uptake is limited (Sect. 5.4). In addition, the number of publicly available tools is extremely small. As the first step to improve this situation, *we suggest that NLP4RE researchers deposit their tools (including the 130 tools identified in this mapping study) in shared repositories* by using, for example, GitHub, as done, among others, by the RE Lab of the University of Utrecht[13]. We further suggest that the shared NLP4RE tools can be structured according to their tasks and targeted RE phases, as shown in Figure 10. While it is important to share data and to build data repositories such as PROMISE[14], we believe building a shared repository of NLP4RE tools is also extremely crucial, especially in a context such as RE in which data are often confidential and difficult to share. A shared tool repository can help researchers develop their work on top of their peers' and would also facilitate technology transfer with industries, as *companies often want to see a working tool before they can be convinced about the feasibility of a collaboration*, for example to adapt the tool to the company context.

[13] https://github.com/RELabUU/

[14] http://promise.site.uottawa.ca/SERepository/datasets-page.html

The use of NLP technologies (RQ5): A large number of NLP technologies (140 NLP techniques, 66 NLP tools and 25 language resources) are identified in the mapping study (Sect. 5.5), but these technologies have not been utilized to full potential, with only a quarter of them being used more frequently. The number of NLP resources used by NLP4RE studies is even smaller in comparison with the number of NLP techniques and tools, with WordNet as the only predominant resource. Our mapping study reveals a clear lack of RE specific resources, such as annotated requirements corpora and benchmark data, as apart from MODIS and CM-1, the remaining corpora, such as British National Corpus and Wikipedia Corpus, are for general NLP applications. The shortage of RE specific resources makes it impossible to develop effective NLP4RE approaches, as using general-purpose language resources to train domain specific language models for processing requirements text would inevitably lead to unreliable results. Also, as we found in another survey [50], the RE community uses token-level metrics such as precision, recall and F1-score to measure the performance of NLP4RE approaches, and these metrics may not be adequate for measuring NLP tasks that involve human judgment or the human-in-the-loop [112]. New metrics begin to emerge to explicitly capture how well the rationales provided by NLP models align with human rationales, and also how faithful these rationales are in influencing models' predictions [113]. The application of different performance metrics to NLP4RE approaches can be a potential collaboration opportunity between the NLP4RE community and the NLP community. Therefore, alongside our suggestion for creating a shared tool repository, *we also suggest that NLP4RE researchers share RE specific datasets, benchmark data and performance metrics*.

In summary, the action of NLPRE researchers in the past years has been successful in providing technical solutions that sufficiently cover the RE process, and it is now time to improve the empirical maturity of the field, with case studies, benchmarks and rigorous evaluations that can lead to more robust tools and open to industrial adoption of NLP4RE solutions. At the same time, researchers should not restrain themselves to consolidated NLP practices, but should also look into more advanced NLP techniques, e.g., for semantic and discourse analysis, and for less explored RE-relevant sources such as interviews and user stories, as they are largely used in practice [10]. Integration between NL and other data, such as RE-related images, vocal and biofeedback data [114] by means of multi-modal analysis [115] is also a challenging and fascinating research avenue.

Suggestions for practitioners: We hope that this mapping study can be useful for practitioners interested in leveraging NLP4RE research results. Specifically, practitioners looking for NLP4RE literature should look outside typical SE dissemination fora, as most NLP4RE studies are published in RE venues, as Figure 4 shows. Practitioners need to be aware that many approaches that they may consider adopting are only lab assessed and most of them were not evaluated with human subjects (Table 7). To help change this situation and move NLP4RE research results to a stronger footing where they could be adopted and accepted for practice, practitioners need to provide researchers with the opportunity to carry out case studies with their involvement. While helping research evaluation will incur some overhead in terms of time and effort, exclusive relationships with the research team may lead to early adoption by the company and a competitive advantage. Practitioners interested in NLP4RE tools may find the dendogram in Figure 10 particularly useful, as it shows what tools have been developed for which tasks and which RE phases. Finally, the set of top NLP technologies, including techniques, tools and resources, identified in Section 5.5, may be particularly useful to practitioners who wish to develop *in-house* NLP4RE tools for their needs, by leveraging existing platforms (Stanford CoreNLP, GATE, NLTK, Open NLP). The top techniques, tools and resources represent the basic, well-established, NLP technologies that are needed to practice NLP4RE. The top 32 NLP techniques identified (POS Tagging, Parsing, etc.) can be used to address the whole set of NLP4RE tasks. Thus, this mapping study provides practitioners with a clear indication of the knowledge needed to develop NLP4RE tools and can be useful to identify the skills required during recruitment of personnel that may be dedicated to the *in-house* development of NLP4RE tools.

7 Study Validity and Limitations

The main threat to the validity of any type of literature review is the question of reliability [116]: if two different studies follow the same research procedures, will they produce the same results [117]? For systematic literature reviews and systematic mapping studies, the threat of reliability can be manifested in the entire review process, from identification of the literature to selection of the relevant studies to the final analysis. To mitigate this validity threat, we took some preventive measures in each step of the study process, described as follows.

Reliability of literature search: Due to the constraints on resources, time and search engines, it is almost impossible to find the entire population of *all* the relevant studies on NLP4RE [116]. To ensure we found as many relevant studies as possible and as close to the actual population as possible, we followed the recommended guidelines [118], [119] to identify the literature (Sect. 4.2.1), formulate the search terms (Sect. 4.2.2) and perform the search (Sect. 4.2.3). However, it is possible that we did not find studies whose authors used other terms that were not included in our search terms. This threat is quite relevant, as RE has a large scope, and some studies may not explicitly refer to RE or use the term "requirement", although they are dealing with requirements activities. We mitigated this problem through initial and targeted searches. Furthermore, this is a mapping study oriented to give an overview of the field, and we argue that the absence of few studies does not jeopardize the statistics. As our main search phase relied on the search engines provided by our chosen libraries, the quality of the search engines could have influenced the completeness of the identified primary studies, as reported by many other systematic reviewers [120], [70].

Reliability of study selection: To ensure our study selection to be as accurate as possible, as free from researcher bias and human errors as possible, we followed a rigorous study selection process, guided by the carefully designed inclusion and exclusion criteria, and enforced by crosschecking and independent checking of the selected and deselected studies (Sect. 4.2.4). We paid a special attention to the last two stages of study selection (Stages 3-4 in Figure 1), because it involved determining the relevance of the remaining studies based on individual interpretations. To overcome the inherent threat arising from this subjectivity, as detailed in Sect. 4.2.4, we followed the recommendation of Kuhrmann et al. [67] by using the majority voting procedure to determine each included or excluded study. Our post hoc calculation shows that between the data inspectors and the lead supervisor, we reached an average agreement of $\kappa = 0.93$ on the final set of selected studies, measured by Cohen's kappa coefficient (κ) [121], suggesting an almost perfect agreement. We therefore believe that the study population we identified is close to the actual population and is a good representative sample of the current state of NLP4RE research.

Reliability of data extraction and classification: To ensure we extract the required data and organize the selected studies accurately, consistently and uniformly, we followed a faceted classification scheme with a comprehensive set of predefined categories (Sect. 4.3). However, the classification scheme was not foolproof for data extraction, as this process involved subjective interpretations and decisions by the researchers. Lack of sufficient details about the design and execution of the reported studies often hindered data extraction. A particular problem arising from identifying exact NLP technologies from the studies was the lack of precise, explicit and standard description of these technologies in the reported studies. For example, when a study stated that it used a simple syntactic technique to analyze a document, it could mean POS tagging only or both POS tagging and parsing. Worse still, some studies stated that they performed a tokenization task, but did not say which NLP tools were used to perform this task. To mitigate this problem, we compiled our own in-house NLP dictionary with a list of NLP techniques, NLP tools and NLP resources. This dictionary was then used to guide us in extracting NLP technologies from the selected studies. The process of classifying the various aspects of the selected studies (such as research types, evaluation methods, RE phases, NLP4RE tasks) also involved subjective decisions by the researchers. To minimize human errors, we carried out regular checks on each category. Whenever there was doubt about the classification of a particular study, we would re-assess that study, re-extract the data and re-classify the data if necessary.

Reliability of data synthesis, analysis and visualization: To ensure the mapping results were as accurate and error-free as possible, we carefully carried out thematic synthesis, descriptive analysis and frequency counting on the extracted data. Thematic synthesis involved standardizing the names of NLP techniques and establishing the types of input document. To synthesize the extracted NLP techniques, we used our NLP dictionary to normalize the names of NLP techniques or combine similar techniques into one. When we discovered new techniques, we also added them to our dictionary. To synthesize input documents, we relied on our knowledge to identify their common types. The synthesized results were reviewed several times and revisions were made to make them as accurate as possible.

8 Conclusion

This article has reported the first-ever systematic mapping study on the landscape of NLP4RE research. From 11,540 search results, 404 primary studies were included in the mapping study and systematically reviewed according to five research questions. These questions interrogate the selected studies to understand their publication status, state of empirical research, research focus, state of tool development, and finally, their usage of NLP technologies. The answers to the research questions show that:

- NLP4RE is an active and thriving research area in RE, which has amassed a large number of publications and attracted widespread attention from diverse communities.
- Most NLP4RE studies (67.08%) are solution proposals that were evaluated using a laboratory experiment or an example application, while only 7.18% of the studies were evaluated in an industrial setting, which highlights a general lack of industrial evaluation of NLP4RE research results.
- The biggest proportion (42.70%) of the NLP4RE studies focuses on the analysis phase, with quality defect detection as their central linguistic analysis task and requirements specification as their commonly processed document type, indicating the current focus of NLP4RE research.
- A total of 130 new tools were proposed to support a range of linguistic analysis tasks, but only 17 of these tools (13.08%) are available for download.
- 231 different NLP technologies comprising 140 techniques, 66 tools and 25 resources were used to support NLP4RE research, but only a quarter of them are used frequently and most popular NLP technologies are lexical or syntactic ones such as POS taggers, syntactic parsers and WordNet.

These findings reveal a huge discrepancy between the state of the art and the state of the practice in current NLP4RE research, indicated by insufficient industrial evaluation of NLP4RE research, little evidence of industrial adoption of the proposed tools, the lack of shared RE-specific language resources, and the lack of NLP expertise in NLP4RE research to advise on the choice of NLP technologies. The findings also carry many important implications for NLP4RE research and practice. We believe that these implications can be used to drive the NLP4RE research agenda. At the top of this agenda there should be concrete plans for active collaboration with practitioners to jointly develop and assess NLP4RE tools, for close collaboration with NLP experts to learn and use cutting-edge NLP technologies for semantic and discourse analysis, and for developing open NLP4RE tools, shared datasets, benchmark data and performance metrics for research evaluation.

In spite of the gaps and limitations, this mapping study also shows that NLP4RE research has made a tremendous progress over the past 15 years, particularly in the areas of publication and tool development. Additionally, recent work in analyzing more challenging documents such as user feedback and legal documents indicates that NLP4RE research has entered a new chapter by taking on more challenging tasks. Furthermore, we noticed that industries are also beginning to leverage NLP4RE research results to develop NLP tools for RE. There is now a real buzz of excitement that NLP4RE research can soon be transformed into a practical technology to support RE practice, especially thanks to novel techniques for transfer learning [97] that can solve the paramount problem of the lack of RE-specific resources.

The mapping results can benefit researchers and practitioners in many ways. For researchers, the selected studies, their deep analysis and categorization can serve as useful references for further research; the identified gaps provide opportunities for innovative research; the identified publication venues, particularly those 12 leading venues, help narrow the search space for literature review in this area or for publishing relevant work. Practitioners interested in tool development can explore the set of the identified NLP4RE research tools and seek potential collaboration with the tool proposers. The identified NLP technologies, together with their usage and their relationships with the NLP4RE tasks, serve as a conceptual framework to help both practitioners and researchers gain a better understanding of what NLP technologies are in use in RE and how they are related. To conclude, we believe this mapping study is an important contribution to the field of NLP4RE, as it provides a substantive, thorough and sophisticated overview of NLP4RE research, and offers many important takeaways for future research and development in this area.

Acknowledgement

We are grateful for the three anonymous reviewers for their detailed, expert comments and thoughtful suggestions; we wish to thank the Editor-in-Chief and the Associate Editor for supporting our manuscript, and Fabiano Dalpiaz and Federica Sarro for providing feedback on previous versions of the manuscript.

References

<bib id="bib1"><number>[1]</number>C. Rolland and C. Proix, "A natural language approach for requirements engineering," in *Advanced information systems engineering*: Springer, 1992, pp. 257-277.</bib>
<bib id="bib2"><number>[2]</number>K. Ryan, "The role of natural language in requirements engineering," in *IEEE International Symposium on Requirements Engineering*, 1993 1993: IEEE, pp. 240-242.</bib>
<bib id="bib3"><number>[3]</number>R. J. Abbott and D. K. Moorhead, "Software requirements and specifications: A survey of needs and languages," *Journal of Systems and Software,* vol. 2, no. 4, pp. 297-316, 1981, doi: http://dx.doi.org/10.1016/0164-1212(81)90004-2.</bib>
<bib id="bib4"><number>[4]</number>L. Mich, F. Mariangela, and N. I. Pierluigi, "Market research for requirements analysis using linguistic tools," *Requirements Engineering,* vol. 9, no. 1, pp. 40-56, 2004.</bib>
<bib id="bib5"><number>[5]</number>M. Kassab, C. Neill, and P. Laplante, "State of practice in requirements engineering: contemporary data," *Innovations in Systems and Software Engineering,* vol. 10, no. 4, pp. 235-241, 2014, doi: 10.1007/s11334-014-0232-4.</bib>
<bib id="bib6"><number>[6]</number>J. Hirschberg and C. D. Manning, "Advances in natural language processing," *Science,* vol. 349, no. 6245, pp. 261-266, 2015.</bib>
<bib id="bib7"><number>[7]</number>G. Goth, "Deep or shallow, NLP is breaking out," *Communications of the ACM,* vol. 59, no. 3, pp. 13-16, 2016.</bib>
<bib id="bib8"><number>[8]</number>A. Ferrari, F. Dell'Orletta, A. Esuli, V. Gervasi, and S. Gnesi, "Natural Language Requirements Processing: A 4D Vision," *IEEE Software,* vol. 34, no. 6, pp. 28-35, 2017, doi: 10.1109/MS.2017.4121207.</bib>
<bib id="bib9"><number>[9]</number>F. Dalpiaz, A. Ferrari, X. Franch, and C. Palomares, "Natural Language Processing for Requirements Engineering: The Best Is Yet to Come," *IEEE Software,* vol. 35, no. 5, pp. 115-119, 2018.</bib>
<bib id="bib10"><number>[10]</number>D. M. Fernández *et al.*, "Naming the Pain in Requirements Engineering," *Empirical Software Engineering,* vol. 22, no. 5, pp. 2298-2338, 2017/10/01 2017, doi: 10.1007/s10664-016-9451-7.</bib>
<bib id="bib11"><number>[11]</number>P. P.-S. Chen, "English sentence structure and entity-relationship diagrams," *Information Sciences,* vol. 29, no. 2, pp. 127-149, 1983.</bib>
<bib id="bib12"><number>[12]</number>R. J. Abbott, "Program design by informal English descriptions," *Communications of the ACM,* vol. 26, no. 11, pp. 882-894, 1983.</bib>
<bib id="bib13"><number>[13]</number>C. Aguilera and D. M. Berry, "The use of a repeated phrase finder in requirements extraction," *Journal of Systems and Software,* vol. 13, no. 3, pp. 209-230, 1990.</bib>
<bib id="bib14"><number>[14]</number>L. Goldin and D. Berry, "AbstFinder, a prototype abstraction finder for natural language text for use in requirements elicitation: design, methodology, and evaluation," *Automated Software Engineering,* vol. 4, no. 4, pp. 375-412., 1994, doi: 10.1109/ICRE.1994.292399.</bib>
<bib id="bib15"><number>[15]</number>L. Mich, "NL-OOPS: from natural language to object oriented requirements using the natural language processing system LOLITA," *Natural language engineering,* vol. 2, no. 02, pp. 161-187, 1996.</bib>
<bib id="bib16"><number>[16]</number>V. Ambriola and V. Gervasi, "Processing natural language requirements," in *The 12th IEEE International Conference on Automated Software Engineering (ASE)*, 1997 1997: IEEE, pp. 36-45.</bib>
<bib id="bib17"><number>[17]</number>H. M. Harmain and R. Gaizauskas, "CM-Builder: An automated NL-based CASE tool," in *The Fifteenth IEEE International Conference on Automated Software Engineering (ASE)*, 2000 2000: IEEE, pp. 45-53.</bib>
<bib id="bib18"><number>[18]</number>F. Fabbrini, M. Fusani, S. Gnesi, and G. Lami, "An automatic quality evaluation for natural language requirements," in *Proceedings of the Seventh International Workshop on Requirements Engineering: Foundation for Software Quality REFSQ*, 2001 2001, vol. 1, pp. 4-5.</bib>
<bib id="bib19"><number>[19]</number>W. M. Wilson, L. H. Rosenberg, and L. E. Hyatt, "Automated analysis of requirement specifications," in *Proceedings of the 19th international conference on Software engineering*, 1997 1997: ACM, pp. 161-171.</bib>
<bib id="bib20"><number>[20]</number>S. F. Tjong and D. M. Berry, "The Design of SREE—A Prototype Potential Ambiguity Finder for Requirements Specifications and Lessons Learned," in *International Working Conference on Requirements Engineering: Foundation for Software Quality*, 2013 2013: Springer, pp. 80-95.</bib>
<bib id="bib21"><number>[21]</number>T. Yue, L. Briand, and Y. Labiche, "aToucan: An Automated Framework to Derive UML Analysis Models from Use Case Models," *ACM Transactions on Software Engineering and Methodology (TOSEM),* vol. 24, no. 3, pp. 1-52, 2015, doi: 10.1145/2699697.</bib>
<bib id="bib22"><number>[22]</number>A. Casamayor, D. Godoy, and M. Campo, "Identification of non-functional requirements in textual specifications: A semi-supervised learning approach," *Information and Software Technology,* vol. 52, no. 4, pp. 436-445, 2010, doi: *https://doi.org/10.1016/j.infsof.2009.10.010*.</bib>
<bib id="bib23"><number>[23]</number>A. Ferrari *et al.*, "Detecting requirements defects with NLP patterns: an industrial experience in the railway domain," *Empir Software Eng,* pp. 1-50, 2018.</bib>
<bib id="bib24"><number>[24]</number>H. Femmer, D. M. Fernández, S. Wagner, and S. Eder, "Rapid quality assurance with requirements smells," *Journal of Systems and Software,* vol. 123, pp. 190-213, 2017.</bib>

<bib id="bib25"><number>[25]</number>D. Falessi, G. Cantone, and G. Canfora, "Empirical principles and an industrial case study in retrieving equivalent requirements via natural language processing techniques," *IEEE Transactions on Software Engineering,* vol. 39, no. 1, pp. 18-44, 2011.</bib>

<bib id="bib26"><number>[26]</number>C. Arora, M. Sabetzadeh, L. Briand, and F. Zimmer, "Automated extraction and clustering of requirements glossary terms," *IEEE Transactions on Software Engineering,* no. 10, pp. 918-945, 2017.</bib>

<bib id="bib27"><number>[27]</number>J. Guo, J. Cheng, and J. Cleland-Huang, "Semantically enhanced software traceability using deep learning techniques," in *IEEE/ACM 39th International Conference on Software Engineering (ICSE)*, 2017 2017: IEEE, pp. 3-14.</bib>

<bib id="bib28"><number>[28]</number>W. Maalej and H. Nabil, "Bug report, feature request, or simply praise? on automatically classifying app reviews," in *IEEE 23rd international requirements engineering conference (RE)*, 2015 2015: IEEE, pp. 116-125.</bib>

<bib id="bib29"><number>[29]</number>E. Guzman, M. Ibrahim, and M. Glinz, "A little bird told me: Mining tweets for requirements and software evolution," in *IEEE International Requirements Engineering Conference (RE)*, 2017 2017: IEEE, pp. 11-20.</bib>

<bib id="bib30"><number>[30]</number>M. Robeer, G. Lucassen, J. M. E. M. van der Werf, F. Dalpiaz, and S. Brinkkemper, "Automated extraction of conceptual models from user stories via NLP," in *IEEE 24th International Requirements Engineering Conference (RE)*, 2016 2016: IEEE, pp. 196-205.</bib>

<bib id="bib31"><number>[31]</number>T. Breaux and A. Antón, "Analyzing regulatory rules for privacy and security requirements," *IEEE transactions on software engineering,* vol. 34, no. 1, pp. 5-20, 2008.</bib>

<bib id="bib32"><number>[32]</number>D. N. Boote and P. Beile, "Scholars before researchers: On the centrality of the dissertation literature review in research preparation," *Educational researcher,* vol. 34, no. 6, pp. 3-15, 2005.</bib>

<bib id="bib33"><number>[33]</number>E. D. Liddy, "Natural Language Processing," in *Encyclopedia of Library and Information Science,*, 2nd ed. New York: Marcel Decker, Inc., 2001.</bib>

<bib id="bib34"><number>[34]</number>G. G. Chowdhury, "Natural language processing," *Annual review of information science and technology,* vol. 37, no. 1, pp. 51-89, 2003.</bib>

<bib id="bib35"><number>[35]</number>P. M. Nadkarni, L. Ohno-Machado, and W. W. Chapman, "Natural language processing: an introduction," *Journal of the American Medical Informatics Association,* vol. 18, no. 5, pp. 544-551, 2011, doi: 10.1136/amiajnl-2011-000464.</bib>

<bib id="bib36"><number>[36]</number>E. Cambria and B. White, "Jumping NLP Curves: A Review of Natural Language Processing Research," *IEEE Computational Intelligence Magazine,* vol. 9, no. 2, pp. 48-57, 2014.</bib>

<bib id="bib37"><number>[37]</number>C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky, "The Stanford CoreNLP Natural Language Processing Toolkit," 2014 2014, pp. 55-60.</bib>

<bib id="bib38"><number>[38]</number>T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent Trends in Deep Learning Based Natural Language Processing," *IEEE Computational Intelligence Magazine,* vol. 13, no. 3, pp. 55-75, 2018, doi: 10.1109/MCI.2018.2840738.</bib>

<bib id="bib39"><number>[39]</number>Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," *nature,* vol. 521, no. 7553, pp. 436-444, 2015.</bib>

<bib id="bib40"><number>[40]</number>P. Sawyer, P. Rayson, and K. Cosh, "Shallow knowledge as an aid to deep understanding in early phase requirements engineering," *Software Engineering, IEEE Transactions on,* vol. 31, no. 11, pp. 969-981, 2005.</bib>

<bib id="bib41"><number>[41]</number>D. Berry, R. Gacitua, P. Sawyer, and S. F. Tjong, "The case for dumb requirements engineering tools," in *International Working Conference on Requirements Engineering: Foundation for Software Quality*, 2012 2012: Springer, pp. 211-217.</bib>

<bib id="bib42"><number>[42]</number>J. Cleland-Huang, R. Settimi, X. Zou, and P. Solc, "Automated classification of non-functional requirements," *Requirements Engineering,* vol. 12, no. 2, pp. 103-120, 2007.</bib>

<bib id="bib43"><number>[43]</number>M. Daneva, D. Damian, A. Marchetto, and O. Pastor, "Empirical research methodologies and studies in Requirements Engineering: How far did we come?," *Journal of systems and software,* vol. 95, pp. 1-9, 2014.</bib>

<bib id="bib44"><number>[44]</number>G. Loniewski, E. Insfran, and S. Abrahão, "A systematic review of the use of requirements engineering techniques in model-driven development," in *International Conference on Model Driven Engineering Languages and Systems*, 2010 2010: Springer, pp. 213-227.</bib>

<bib id="bib45"><number>[45]</number>T. Yue, L. C. Briand, and Y. Labiche, "A systematic review of transformation approaches between user requirements and analysis models," *Requirements engineering,* vol. 16, no. 2, pp. 75-99, 2011.</bib>

<bib id="bib46"><number>[46]</number>J. Nicolás and A. Toval, "On the generation of requirements specifications from software engineering models: A systematic literature review," *Information and Software Technology,* vol. 51, no. 9, pp. 1291-1307, 2009.</bib>

<bib id="bib47"><number>[47]</number>D. Dermeval *et al.*, "Applications of ontologies in requirements engineering: a systematic review of the literature," *Requirements Engineering,* vol. 21, no. 4, pp. 405-437, 2016.</bib>

<bib id="bib48"><number>[48]</number>M. Irshad, K. Petersen, and S. Poulding, "A systematic literature review of software requirements reuse approaches," *Information and Software Technology,* vol. 93, pp. 223-245, 2018.</bib>

<bib id="bib49"><number>[49]</number>R. Torkar, T. Gorschek, R. Feldt, M. Svahnberg, U. A. Raja, and K. Kamran, "Requirements traceability: a systematic review and industry case study," *International Journal of Software Engineering and Knowledge Engineering,* vol. 22, no. 03, pp. 385-433, 2012.</bib>

<bib id="bib50"><number>[50]</number>M. Binkhonain and L. Zhao, "A Review of Machine Learning Algorithms for Identification and Classification of Non-Functional Requirements," *Expert Systems with Applications: X,* vol. 1, pp. 1 - 13, 2019.</bib>

<bib id="bib51"><number>[51]</number>N. H. Bakar, Z. M. Kasirun, and N. Salleh, "Feature extraction approaches from natural language requirements for reuse in software product lines: A systematic literature review," *Journal of Systems and Software,* vol. 106, pp. 132-149, 2015, doi: *http://dx.doi.org/10.1016/j.jss.2015.05.006*.</bib>

<bib id="bib52"><number>[52]</number>Y. Li, S. Schulze, and G. Saake, "Reverse engineering variability from natural language documents: A systematic literature review," in *The 21st International Systems and Software Product Line Conference*, 2017 2017: ACM, pp. 133-142.</bib>

<bib id="bib53"><number>[53]</number>W. Martin, F. Sarro, Y. Jia, Y. Zhang, and M. Harman, "A survey of app store analysis for software engineering," *IEEE transactions on software engineering,* vol. 43, no. 9, pp. 817-847, 2016.</bib>

<bib id="bib54"><number>[54]</number>M. Tavakoli, L. Zhao, A. Heydari, and G. Nenadić, "Extracting useful software development information from mobile application reviews: A survey of intelligent mining techniques and tools," *Expert Systems with Applications,* vol. 113, pp. 189 - 199, 2018.</bib>

<bib id="bib55"><number>[55]</number>E. C. G. R. Santos, K. Villela, "A Taxonomy for User Feedback Classifications," in *International Workshop on Natural Language for Requirements Engineering (NLP4RE)*, 2019, pp. 1 - 10.</bib>

<bib id="bib56"><number>[56]</number>E. C. G. R. Santos, K. Villela, "An Overview of User Feedback Classification Approaches," in *International Workshop on Natural Language for Requirements Engineering (NLP4RE)*, 2019, pp. 1 - 10.</bib>

<bib id="bib57"><number>[57]</number>N. Nazar, Y. Hu, and H. Jiang, "Summarizing software artifacts: A literature review," *Journal of Computer Science and Technology,* vol. 31, no. 5, pp. 883-909, 2016.</bib>

<bib id="bib58"><number>[58]</number>I. Ahsan, W. H. Butt, M. A. Ahmed, and M. W. Anwar, "A comprehensive investigation of natural language processing techniques and tools to generate automated test cases," in *The Second International Conference on Internet of things, Data and Cloud Computing*, 2017 2017, pp. 1-10.</bib>

<bib id="bib59"><number>[59]</number>V. Garousi, S. Bauer, and M. Felderer, "NLP-assisted software testing: A systematic mapping of the literature," *Information and Software Technology,* vol. 126, 2020, doi: *https://doi.org/10.1016/j.infsof.2020.106321*.</bib>

<bib id="bib60"><number>[60]</number>U. S. Shah and D. C. Jinwala, "Resolving ambiguities in natural language software requirements: a comprehensive survey," *ACM SIGSOFT Software Engineering Notes,* vol. 40, no. 5, pp. 1-7, 2015.</bib>

<bib id="bib61"><number>[61]</number>A. Casamayor, D. Godoy, and M. Campo, "Mining textual requirements to assist architectural software design: a state of the art review," *Artificial Intelligence Review,* vol. 38, no. 3, pp. 173-191, 2012.</bib>

<bib id="bib62"><number>[62]</number>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 *International conference on information science and applications*, 2017 2017: Springer, pp. 485-493.</bib>

<bib id="bib63"><number>[63]</number>D. Janssens, "Natural Language Processing in Requirements Elicitation and Requirements Analysis: a Systematic Literature Review," Master of Science, Department of Information and Computing Sciences, Utrecht University, 2019. [Online]. Available: *https://dspace.library.uu.nl/handle/1874/380338</bib>*

<bib id="bib64"><number>[64]</number>C. L. Perryman, "Mapping studies," *Journal of the Medical Library Association : JMLA,* vol. 104, no. 1, pp. 79-82, 2016, doi: 10.3163/1536-5050.104.1.014.</bib>

<bib id="bib65"><number>[65]</number>K. Petersen, S. Vakkalanka, and L. Kuzniarz, "Guidelines for conducting systematic mapping studies in software engineering: An update," *Information and Software Technology,* vol. 64, pp. 1-18, 2015.</bib>

<bib id="bib66"><number>[66]</number>B. A. Kitchenham, D. Budgen, and P. Brereton, *Evidence-Based Software Engineering and Systematic Reviews*. CRC press, 2015.</bib>

<bib id="bib67"><number>[67]</number>M. Kuhrmann, D. M. Fernández, and M. Daneva, "On the pragmatic design of literature studies in software engineering: an experience-based guideline," *Empirical software engineering,* vol. 22, no. 6, pp. 2852-2891, 2017.</bib>

<bib id="bib68"><number>[68]</number>B. A. Kitchenham, D. Budgen, and O. P. Brereton, "Using mapping studies as the basis for further research – A participant - observer case study," *Information and Software Technology,* vol. 53, no. 6, pp. 638-651, 2011, doi: *http://dx.doi.org/10.1016/j.infsof.2010.12.011*.</bib>

<bib id="bib69"><number>[69]</number>R. Wieringa, N. Maiden, N. Mead, and C. Rolland, "Requirements engineering paper classification and evaluation criteria: a proposal and a discussion," *Requirements Engineering,* vol. 11, no. 1, pp. 102-107, 2006.</bib>

<bib id="bib70"><number>[70]</number>L. Chen and M. A. Babar, "A systematic review of evaluation of variability management approaches in software product lines," *Information and Software Technology,* vol. 53, no. 4, pp. 344-362, 2011.</bib>

<bib id="bib71"><number>[71]</number>B. H. C. Cheng and J. M. Atlee, "Research directions in requirements engineering," in *Requirements Engineering Conference*, 2007 2009: IEEE Computer Society, pp. 285-303.</bib>

<bib id="bib72"><number>[72]</number>J. Cleland-Huang, C. K. Chang, G. Sethi, K. Javvaji, H. Hu, and J. Xia, "Automating speculative queries through event-based requirements traceability," in *IEEE Joint International Conference on Requirements Engineering*, 2002 2002: IEEE, pp. 289-296.</bib>

<bib id="bib73"><number>[73]</number>J. H. Hayes, A. Dekhtyar, and S. K. Sundaram, "Advancing candidate link generation for requirements tracing: The study of methods," *IEEE Transactions on Software Engineering,* vol. 32, no. 1, pp. 4-19, 2006.</bib>

<bib id="bib74"><number>[74]</number>L. Kof, R. Gacitua, M. Rouncefield, and P. Sawyer, "Concept mapping as a means of requirements tracing," in *Third International Workshop on Managing Requirements Knowledge (MARK)*, 2010 2010: IEEE, pp. 22-31.</bib>

<bib id="bib75"><number>[75]</number>H. Sultanov and J. H. Hayes, "Application of reinforcement learning to requirements engineering: requirements tracing," in*2013 21st IEEE International Requirements Engineering Conference (RE)*, 15-19 July 2013 2013, pp. 52-61, doi: 10.1109/RE.2013.6636705.</bib>

<bib id="bib76"><number>[76]</number>W. Alhoshan, R. T. Batista-Navarro, and L. Zhao, "Semantic Frame Embeddings for Detecting Relations between Software Requirements," in *Proceedings of the 13th International Conference on Computational Semantics*, 2019 2019, pp. 44-51.</bib>

<bib id="bib77"><number>[77]</number>J. N. o. Dag, V. Gervasi, S. Brinkkemper, and B. Regnell, "Speeding up requirements management in a product software company: linking customer wishes to product requirements through linguistic engineering," in *Proceedings. 12th IEEE International Requirements Engineering Conference, 2004.*, 10-10 Sept. 2004 2004, pp. 283-294, doi: 10.1109/ICRE.2004.1335685.</bib>

<bib id="bib78"><number>[78]</number>A. Mahmoud and N. Niu, "An experimental investigation of reusable requirements retrieval," in*2010 IEEE International Conference on Information Reuse & Integration*, 4-6 Aug. 2010 2010, pp. 330-335, doi: 10.1109/IRI.2010.5558914.</bib>

<bib id="bib79"><number>[79]</number>H. Dumitru *et al.*, "On-demand feature recommendations derived from mining public product descriptions," in*2011 33rd International Conference on Software Engineering (ICSE)*, 21-28 May 2011 2011, pp. 181-190, doi: 10.1145/1985793.1985819.</bib>

<bib id="bib80"><number>[80]</number>J. Corbin and A. Strauss, *Basics of qualitative research: Techniques and procedures for developing grounded theory*, 4th ed. Sage publications, 2014.</bib>

<bib id="bib81"><number>[81]</number>H. Priest, P. Roberts, and L. Woods, "An overview of three different approaches to the interpretation of qualitative data. Part 1: Theoretical issues," *Nurse Researcher (through 2013),* vol. 10, no. 1, p. 43, 2002.</bib>

<bib id="bib82"><number>[82]</number>D. S. Cruzes and T. Dyba, "Recommended steps for thematic synthesis in software engineering," in *International Symposium on Empirical Software Engineering and Measurement*, 2011 2011: IEEE, pp. 275-284.</bib>

<bib id="bib83"><number>[83]</number>I. D. Cooper, "What is a "mapping study?"," *Journal of the Medical Library Association : JMLA,* vol. 104, no. 1, pp. 76-78, 2016, doi: 10.3163/1536-5050.104.1.013.</bib>

<bib id="bib84"><number>[84]</number>K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, "Systematic mapping studies in software engineering," in*12th International Conference on Evaluation and Assessment in Software Engineering*, 2008 2008, vol. 17: sn, 1 ed., p. 1 10.</bib>

<bib id="bib85"><number>[85]</number>J. Horkoff *et al.*, "Goal-oriented requirements engineering: an extended systematic mapping study," *Requirements Engineering,* vol. 24, no. 2, pp. 133-160, 2019, doi: 10.1007/s00766-017-0280-z.</bib>

<bib id="bib86"><number>[86]</number>A. Meidan, J. A. García-García, I. Ramos, and M. J. Escalona, "Measuring Software Process: A Systematic Mapping Study," *ACM Computing Surveys (CSUR),* vol. 51, no. 3, pp. 1-32, 2018.</bib>

<bib id="bib87"><number>[87]</number>D. I. K. Sjoberg, T. Dyba, and M. Jorgensen, "The future of empirical methods in software engineering research," in *Future of Software Engineering*, 2007 2007: IEEE Computer Society, pp. 358-378.</bib>

<bib id="bib88"><number>[88]</number>L. Zhao, P. Loucopoulos, E. Kavakli, and J. K. Letsholo, "User Studies on End-User Service Composition: A Literature Review and a Design Framework," *ACM Transactions on the Web,* Journal vol. 13, no. 3, pp. 1- 46, July 2019.</bib>

<bib id="bib89"><number>[89]</number>P. Hider and B. Pymm, "Empirical research methods reported in high-profile LIS journal literature," *Library & information science research,* vol. 30, no. 2, pp. 108-114, 2008.</bib>

<bib id="bib90"><number>[90]</number>D. Falessi *et al.*, "Empirical software engineering experts on the use of students and professionals in experiments," *Empirical Software Engineering,* vol. 23, no. 1, pp. 452-489, 2018, doi: 10.1007/s10664-017-9523-3.</bib>

<bib id="bib91"><number>[91]</number>B. A. Kitchenham *et al.*, "Preliminary Guidelines for Empirical Research in Software Engineering," *IEEE Transactions on Software Engineering,* vol. 28, no. 8, pp. 721-734, 2002.</bib>

<bib id="bib92"><number>[92]</number>L. V. G. Carreño and K. Winbladh, "Analysis of user comments: an approach for software requirements evolution," in*35th international conference on software engineering (ICSE)*, 2013 2013: IEEE, pp. 582-591.</bib>

<bib id="bib93"><number>[93]</number>C. Anderson, *The Long Tail: Why the Future of Business Is Selling Less of More*. Hachette Books, 2006.</bib>

<bib id="bib94"><number>[94]</number>T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781,* 2013.</bib>

<bib id="bib95"><number>[95]</number>I. Iacobacci, M. T. Pilehvar, and R. Navigli, "Embeddings for word sense disambiguation: An evaluation study," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics* 2016 2016, pp. 897-907.</bib>

<bib id="bib96"><number>[96]</number>L. Fleming, "Breakthroughs and The "Long Tail" of Innovation," *MIT Sloan Management Review,* vol. 49, no. 1, pp. 69 - 74, 2007.</bib>

<bib id="bib97"><number>[97]</number>J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805,* 2018.</bib>

<bib id="bib98"><number>[98]</number>T. Hey, J. Keim, A. Koziolek, and W. F. Tichy, "NoRBERT: Transfer learning for requirements classification," in *IEEE 28th International Requirements Engineering Conference (RE)*, 2020 2020: IEEE, pp. 169-179.</bib>

<bib id="bib99"><number>[99]</number>A. Sainani, P. R. Anish, V. Joshi, and S. Ghaisas, "Extracting and Classifying Requirements from Software Engineering Contracts," in *IEEE 28th International Requirements Engineering Conference (RE)*, 2020 2020: IEEE, pp. 147-157.</bib>

<bib id="bib100"><number>[100]</number>J. H. Hayes, A. Dekhtyar, and S. K. Sundaram, "Advancing candidate link generation for requirements tracing: The study of methods," *IEEE Transactions on Software Engineering,* vol. 32, no. 1, p. 4, 2006.</bib>

<bib id="bib101"><number>[101]</number>A. Ferrari, G. O. Spagnolo, and S. Gnesi, "Pure: A dataset of public requirements documents," in *IEEE 25th International Requirements Engineering Conference (RE)*, 2017 2017: IEEE, pp. 502-505.</bib>

<bib id="bib102"><number>[102]</number>W. Alhoshan, R. Batista-Navarro, and L. Zhao, "Towards a corpus of requirements documents enriched with semantic frame annotations," in *IEEE 26th International Requirements Engineering Conference (RE)*, 2018 2018: IEEE, pp. 428-431.</bib>

<bib id="bib103"><number>[103]</number>W. Zogaan, P. Sharma, M. Mirahkorli, and V. Arnaoudova, "Datasets from fifteen years of automated requirements traceability research: Current state, characteristics, and quality," in *IEEE 25th International Requirements Engineering Conference (RE)*, 2017 2017: IEEE, pp. 110-121.</bib>

<bib id="bib104"><number>[104]</number>F. Dalpiaz. Requirements data sets (user stories) </bib>

<bib id="bib105"><number>[105]</number>F. Dalpiaz, I. van der Schalk, S. Brinkkemper, F. B. Aydemir, and G. Lucassen, "Detecting terminological ambiguity in user stories: Tool and experimentation," *Information and Software Technology,* vol. 110, pp. 3-16, 2019/06/01/ 2019, doi: *https://doi.org/10.1016/j.infsof.2018.12.007*.</bib>

<bib id="bib106"><number>[106]</number>P. Runeson, M. Host, A. Rainer, and B. Regnell, *Case study research in software engineering: Guidelines and examples*. John Wiley & Sons, 2012.</bib>

<bib id="bib107"><number>[107]</number>C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in software engineering*. Springer Science & Business Media, 2012.</bib>

<bib id="bib108"><number>[108]</number>C. Manning and H. Schutze, *Foundations of statistical natural language processing*. MIT press, 1999.</bib>

<bib id="bib109"><number>[109]</number>A. Ferrari, G. O. Spagnolo, and F. Dell'Orletta, "Mining commonalities and variabilities from natural language documents," in *Proceedings of the 17th International Software Product Line Conference*, 2013 2013, pp. 116-120.</bib>

<bib id="bib110"><number>[110]</number>A. Al-Subaihin, F. Sarro, S. Black, and L. Capra, "Empirical comparison of text-based mobile apps similarity measurement techniques," *Empirical Software Engineering,* vol. 24, no. 6, pp. 3290-3315, 2019.</bib>

<bib id="bib111"><number>[111]</number>V. Garousi, S. Bauer, and M. Felderer, "NLP-assisted software testing: A systematic mapping of the literature," *Information and Software Technology,* vol. 126, p. 106321, 2020/10/01/ 2020, doi: *https://doi.org/10.1016/j.infsof.2020.106321*.</bib>

<bib id="bib112"><number>[112]</number>A. Ferrari, "Natural language requirements processing: from research to practice," in *IEEE/ACM 40th International Conference on Software Engineering: Companion (ICSE-Companion)*, 2018 2018: IEEE, pp. 536-537.</bib>

<bib id="bib113"><number>[113]</number>J. DeYoung *et al.*, "Eraser: A benchmark to evaluate rationalized nlp models," *arXiv preprint arXiv:1911.03429,* 2019.</bib>

<bib id="bib114"><number>[114]</number>D. Girardi, A. Ferrari, N. Novielli, P. Spoletini, D. Fucci, and T. Huichapa, "The Way it Makes you Feel Predicting Users' Engagement during Interviews with Biofeedback and Supervised Learning," in *The 28th IEEE International Requirements Engineering Conference (RE'20)* 2020: IEEE, pp. 1 - 12.</bib>

<bib id="bib115"><number>[115]</number>M. Nayebi, "Eye of the Mind: Image Processing for Social Coding," in Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER '20), 2020, pp. 49 - 52.</bib>

<bib id="bib116"><number>[116]</number>C. Wohlin, P. Runeson, P. A. d. M. S. Neto, E. Engström, I. do Carmo Machado, and E. S. De Almeida, "On the reliability of mapping studies in software engineering," *Journal of Systems and Software,* vol. 86, no. 10, pp. 2594-2610, 2013.</bib>

<bib id="bib117"><number>[117]</number>R. K. Yin, *Case study research: Design and methods*. Sage publications, 2013.</bib>

<bib id="bib118"><number>[118]</number>B. Kitchenham and e. al., "Guidelines for performing systematic literature reviews in software engineering," in "EBSE Technical Report " Keele University Keele, Staffs, ST5 5BG, UK Technical Report EBSE-2007-01, 9 July 2007 2007.</bib>

<bib id="bib119"><number>[119]</number>B. A. Kitchenham, E. Mendes, and G. H. Travassos, "Cross versus within-company cost estimation studies: A systematic review," *IEEE Transactions on Software Engineering,* vol. 33, no. 5, pp. 316-329, 2007.</bib>

<bib id="bib120"><number>[120]</number>P. Brereton, B. A. Kitchenham, D. Budgen, M. Turner, and M. Khalil, "Lessons from Applying the Systematic Literature Review Process within the Software Engineering domain," *Journal of Systems and Software,* vol. 80, no. 4, pp. 571-583, 2007, doi: *http://dx.doi.org/10.1016/j.jss.2006.07.009*.</bib>

<bib id="bib121"><number>[121]</number>J. R. Landis and G. G. Koch, "The measurement of observer agreement for categorical data," *biometrics,* pp. 159-174, 1977.</bib>