Otto-von-Guericke-University Magdeburg
Faculty of Electrical Engineering and Information Technology
Chair of Integrated Automation

Non-Technical Project

Literature Review on Requirements Extraction using Natural Language Processing

submitted: July 26, 2023

by: Rutvij Ashokbhai Chitroda
born on August 10, 1997
in Surat, Gujarat, India

Advisor: Dipl.-Math. Robin Gröpler

Examiner: Prof. Dr.-Ing. Christian Diedrich

Task of the Non-Technical Project in the Original:

Requirements engineering is a crucial aspect of the software development process, as it lays the foundation for subsequent stages, such as implementation and testing of a software system. However, eliciting and retrieving requirements from natural language documents remains a challenging and time-consuming process that often requires a significant amount of manual work. To address this problem, natural language processing (NLP) methods and tools have been developed to help extract requirements concepts and automate the process of requirements extraction.

This work aims to provide a systematic literature review of the current state of the art in requirements extraction using NLP. The primary focus will be on identifying publications with well-justified and documented NLP approaches for requirements extraction. To achieve this goal, a search strategy and selection of relevant studies based on inclusion and exclusion criteria will be defined. A mapping study will be conducted with a quantitative analysis covering the following aspects:

- Year-wise distribution of publications
- Algorithms and models
- NLP technologies and methodologies
- Used NLP libraries and programming languages
- Countries of the authors who contributed to the publications

Declaration by the candidate

I hereby declare that this report is my own work and effort and that it has not been submitted anywhere for any award. Where other sources of information have been used, they have been marked.

The work has not been presented in the same or a similar form to any other testing authority and has not been made public.

I hereby also entitle a right of use (free of charge, not limited locally and for an indefinite period of time) that my report can be duplicated, saved and archived by the Otto von Guericke University of Magdeburg (OvGU) or any commissioned third party (e. g. *iParadigms Europe Limited*, provider of the plagiarism-detection service "Turnitin") exclusively in order to check it for plagiarism and to optimize the appraisal of results.

Magdeburg, July 26, 2023

Contents

List of Figures

List of Tables

1 Introduction

Requirements engineering is a crucial aspect of the software development process, as it lays the foundation for subsequent stages, such as the implementation and testing of a software system. However, eliciting and retrieving requirements from natural language documents remains a challenging and time-consuming process that often requires a significant amount of manual work. To address this problem, natural language processing (NLP) methods and tools have been developed to help extract requirements concepts and automate the process of requirements extraction.

This work aims to provide a systematic literature review of the current state of the art in requirement extraction using NLP. The primary focus will be on identifying publications with well-justified and documented NLP approaches for requirement extraction. To achieve this goal, a search strategy and selection of relevant studies based on inclusion and exclusion criteria will be defined. The motivation behind writing this work is to identify different NLP techniques, algorithms, and models, libraries and tools, NLP technologies, and authors from different countries who contributed to publishing articles on requirements extraction using NLP, and to showcase all this statistical data in our paper.

Requirements elicitation is the process of gathering and defining the requirements for a software system. The ultimate goal of requirements elicitation is to provide high-quality software that meets the expectations of stakeholders and customers needs. Requirements elicitation helps to make sure that the software development process is clear without any misunderstandings from the developer's side as well as the customer's side. This process is a very crucial part of the software development lifecycle and is generally carried out at the beginning of the project. During the requirements elicitation process, different sources and various documents are used to group the information. Requirements elicitation is perhaps the most difficult, error-prone, and communication-intensive process in software development. It can be successful only through an effective customer-developer partnership. It is necessary to know what the users really need.

This paper is mainly divided into five parts: Citavi (Reference Management Software), Data Collection, Quantitative Analysis, Limitations, and Summary and Conclusion. In the first part, we discuss Citavi, a reference management software that helped us manage all the references used in our literature review. In the second part, we discuss the selection of databases and the method of collecting articles by means of a search expression. Here the method of formulating the search expression, inclusion and exclusion criteria, and their respective outcomes are discussed. In the third part, quantitative analysis is carried out on the basis of the data we have collected using different parameters. In the fourth and fifth parts, limitations and conclusions are discussed.

2 Citavi - Reference Management Software

For a literature review, a structured approach for collecting, organising, and analysing references is very important. Reference management tools play a crucial role in any type of literature review. There are many key advantages to using a reference management tool, like time savings, import and export capabilities, efficient organisation, collaboration, etc. There are many reference management tools for this purpose, like Zotero, Citavi, Mendeley, EndNote, RefWorks, BibTex, and many more. For our work, we have used Citavi.

Why should I use reference management software?

- Collecting citations, references, and files.
- Managing your references in order to maintain an overview even over a longer period of time.
- Display of reference lists and automatic citation.

Advantages of Citavi

- It provides either a German, English, French, Spanish, Italian, or Portuguese user interface.
- In addition to reference management, Citavi offers tools for knowledge organisation and task scheduling.
- Citavi Team offers the opportunity to work on a project in teams.
- Directly, we can import the article from various search engines, like Scopus, in the form of BibTex.

Figure 2.1 shows the Citavi software interface, which we have used for our literature review. We have exported all the documents using BibTex from Scopus to Citavi and further classified them as per our needs in Citavi.

Figure 2.1: Citavi interface

3 Data Collection

An extensive literature survey was performed by Zhao et al. [1], published in 2021, about the entire requirements engineering domain and the use of NLP techniques. This study will serve as a first entry point for the selection of sources that are mentioned in Zhao's paper, "Natural Language Processing for Requirements Engineering: A Systematic Mapping Study".

The research methodology was adopted from Lenis R. Wong et al. [2], published in 2017, "A Systematic Literature Review About Software Requirements Elicitation".

3.1 Selecting the Database

There are many databases available as the main data sources for our literature survey: ACM Digital Library (ACM), Web of Science (WOS), Scopus (Elsevier), Google Scholar, Semantic Scholar, etc. The criteria we used to finalise the database were advanced search, more specifically search based on abstract, keywords, article title, etc.; advanced filtering options; sorting capabilities; and the ability to export the selected articles directly in the form of Bibtex for further processing. The results given by all the search engines were different; e.g., Web of Science gave many irrelevant sources, and Google Scholar also gave a few relevant sources. The major advantage of using Scopus for our literature review was that it gave the most relevant results, and the number of results we got after applying different filters made it very practical for us to study in a short period of time compared to other databases. It can also easily integrate with other research tools and platforms, making it easier to export citations, import references into citation management software like Citavi, and connect with other research databases or discovery services.

3.2 Formulating the Search Expression

Formulating the search expression can be challenging, as it is a completely trial-and-error method because the only data you know are the keywords that you would like to search for in the search engine. In our literature survey, the following stream searches including logical operators (AND, OR) were added for advanced search. The target of our search expression was to get the most recent and updated list of articles so that we could further analyse them. The search was done in a sequential way; initially, a broader list of articles was generated, and afterwards, specific domains were selected for our literature survey. In our literature survey, we have used articles published in the last 5 years (January 2018 - May 2023).

Based on the above considerations, the search string is defined as follows:

```
TITLE-ABS-KEY ("natural language processing" OR NLP) AND
TITLE-ABS-KEY ("requirements extraction" OR "requirements elicitation"
OR "requirements retrieval"
```

The initial keywords used in the search expression were ''nlp" AND "natural language processing" AND "requirements extraction," which gave an initial list of publications, and afterwards adding more keywords such as "requirements elicitation" OR "requirements retrieval" made the result stronger and gave us the required and relevant articles. We also provided the keywords in capital letters, e.g., "NLP" OR "NATURAL LANGUAGE PROCESSING," but the results were the same regardless of capital and small letters.

3.3 Inclusion and Exclusion Criteria for Selecting Relevant Studies

Due to the time restrictions, we have used some strict criteria in our work. In Table 3.1, the major criteria that helped us narrow down the searched documents were the publication dates from 2018 to 2023.

Table 3.1: Inclusion and exclusion criteria

Inclusion(I) criteria	**Exclusion(E) criteria**
The article should be written in English.	Review articles should be excluded.
The full text of the article should be accessible.	Articles not related to Software Engineering should be excluded.
The article should include at least one NLP technique or tool.	If there are multiple similar works from the same author, only one of the earliest works should be used.
The article should have been published between 2018 and 2023.	

3.4 Outcome

Figure 3.1 shows the flowchart of the literature review process. Two data sources were used: Zhao's paper[1] and Scopus for our literature review.

[1] https://github.com/waadalhoshan/NLP4RE/blob/main/Selected%20Studies/Selected-Studies.bib

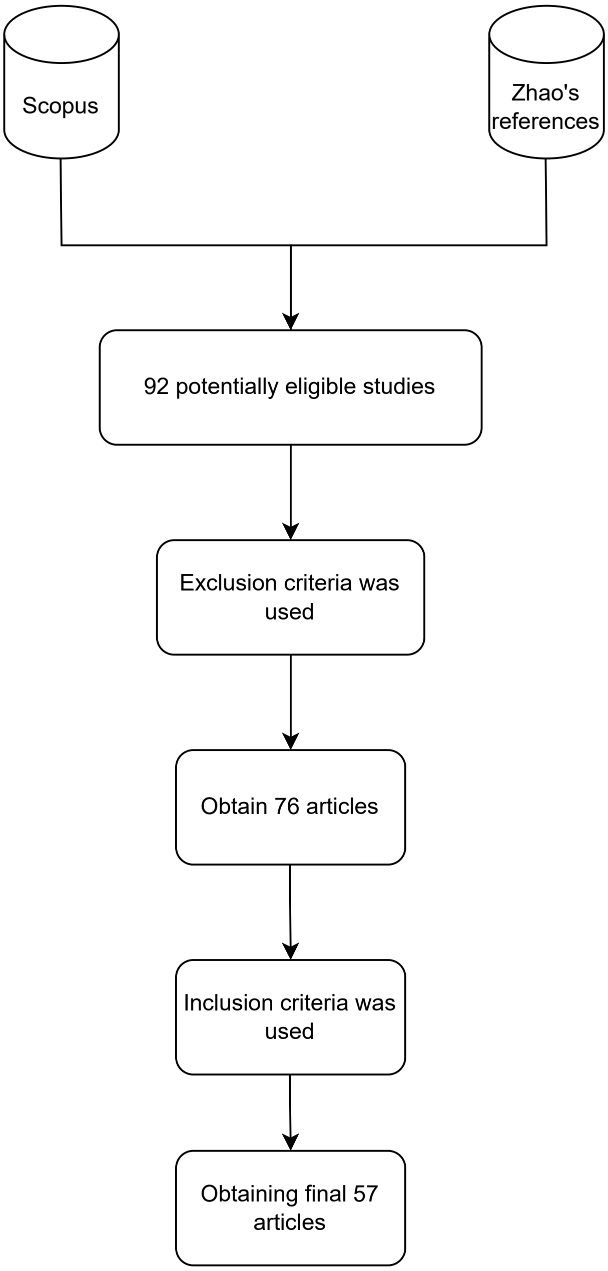

Figure 3.1: Flowchart of the literature review process

4 Quantitative Analysis

Quantitative analysis refers to the process of using mathematical and statistical methods to analyse data and extract meaningful insights. It involves the systematic approach of collecting, organising, and interpreting numerical data to understand patterns, relationships, and trends.

Table 4.1 shows the types of contributions in the requirements extraction. It shows that around 25 different algorithms and models are used 86 times, 2 different programming languages are used 25 times, 15 different libraries and tools are used 54 times, and at last, around 28 different techniques are used 108 times.

Table 4.1: Types of contributions in the requirements extraction

Types of Contributions	**Total number of times used**	**Total number of identified objects**
Algorithms and Models	86	25
Programming Languages	25	2
Libraries and Tools	54	15
Techniques	108	28

4.1 Year-Wise Distribution of Studies

In the number of studies published from 2018 - 2023 in Figure 4.1, we observe that around 21 studies were published in 2018, 11 studies were published in 2019, 4 studies were published in 2020, 10 studies were published in 2021, 9 studies were published in 2022, and 2 studies were published until May 2023. The references taken from Zhao are limited to only two years for our literature review, 2018 and 2019, while Scopus is used for our entire literature review from 2018 to May 2023. So in the Figure 4.1, it is seen that the number of publications published from 2018 to 2019 is much higher because Zhao has done a detailed study behind it. If Zhao could extend their study to 2023, the result would be an increasing number of years passing by. Figure 4.1 shows the slight downward trend in the last few years as the number of studies kept on reducing with passing years with the inclusion and exclusion criteria, while Figure 4.2 shows the upward trend in the last few years without the inclusion and exclusion criteria, which depicts the growing interest in this research field.

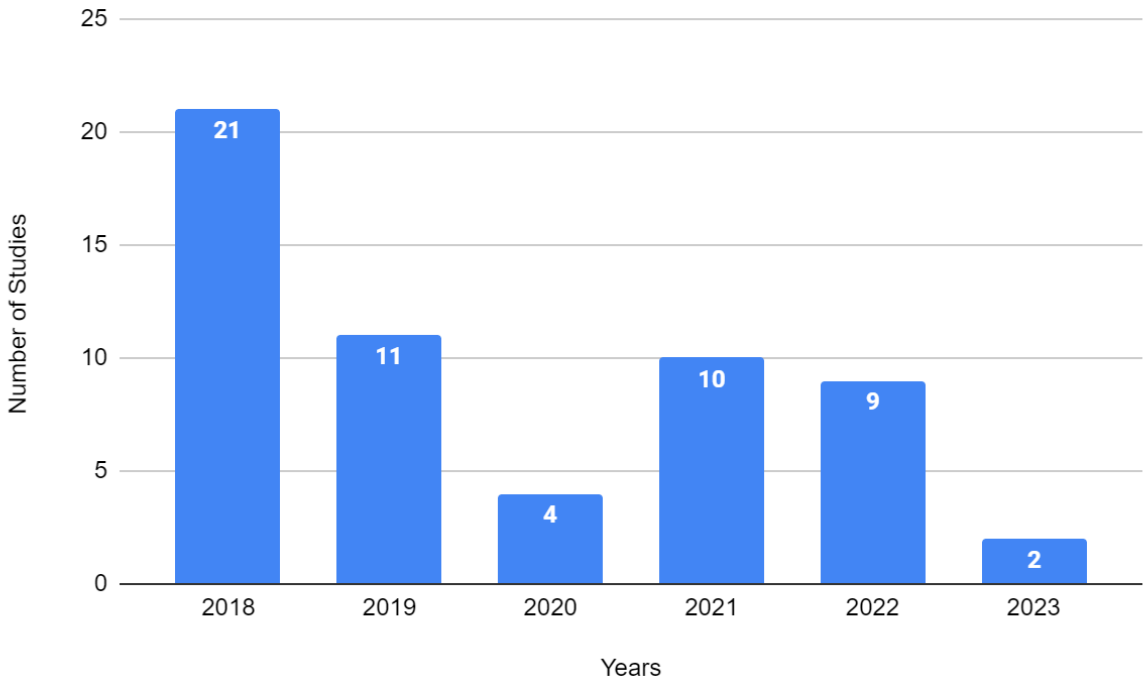

Figure 4.1: Publication timeline of the 57 selected studies

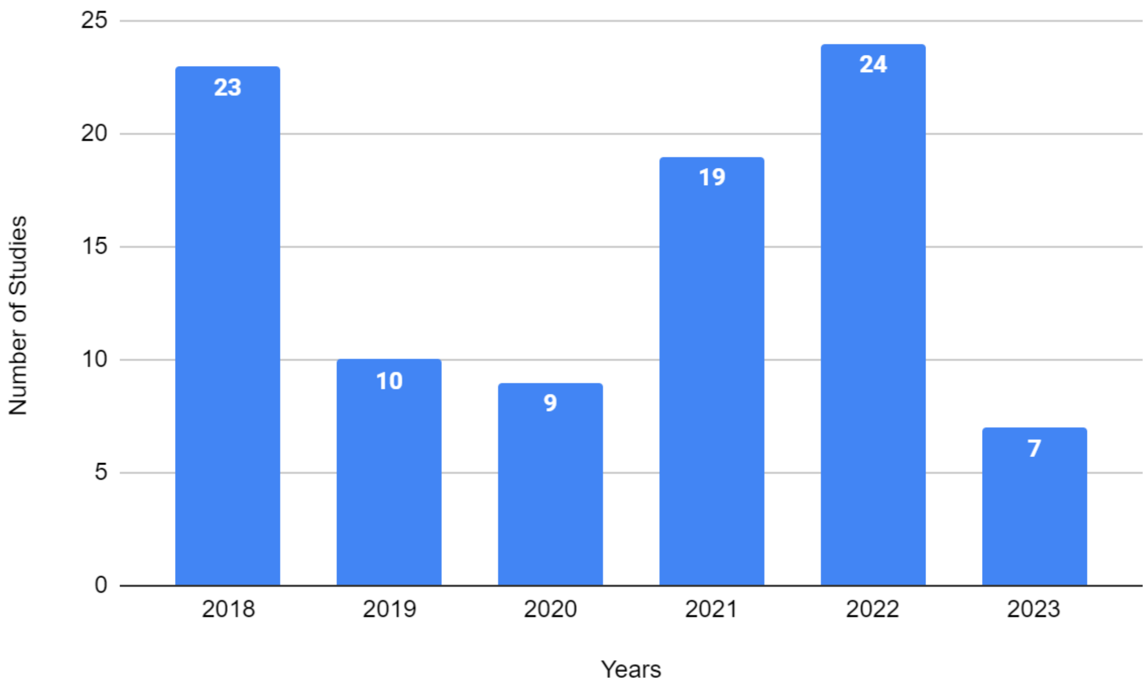

Figure 4.2: Publication timeline of the 92 studies without inclusion and exclusion criteria

4.2 NLP Techniques

Figure 4.3 shows that the most frequently used NLP technique is POS tagging (used by 31 studies), while the next most used techniques are Tokenization (by 17 studies), Lemmatization (by 11 studies), Stop-word removal and Stemming and Named entity recognition (all by 6 studies) and Semantic analysis (by 4 studies).

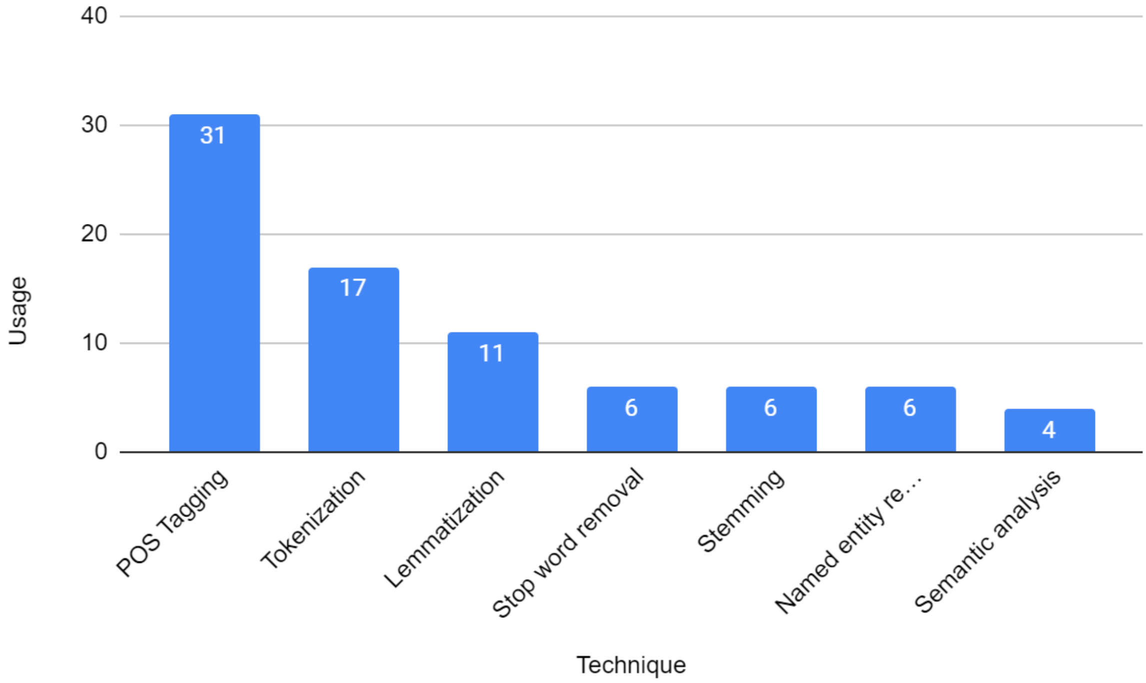

Figure 4.3: Most frequently used NLP techniques

Table 4.2 shows the list of different NLP techniques identified by Zhao [3] and Wikipedia[1]. In total, 27 different techniques with explanations are described in Table 4.2.

Table 4.2: List of NLP techniques

ID	Name	Explanation
1	Regular Expression	A special series of strings for describing a text pattern for the purpose of searching for or replacing the described items.
2	Syntactic trees	A Syntax tree, or parse tree, is a tree representation of different syntactic categories in a sentence. It helps us understand the syntactic structure of a sentence.
3	Pattern matching	Pattern matching is the process of matching tokens or phrases within chunks of text or whole documents.

[1]https://en.wikipedia.org/wiki/Natural_language_processing

4	Template filling approach	Template filling is an efficient approach to extracting and structuring complex information from text. It plays a major role in information extraction (IE) systems (Information Extraction; Text Mining) to merge information across multiple sentences to identify all role fillers of interest.
5	POS Tagging	POS Tagging (or Tagging) processes a sequence of words, and attaches a POS tag to each word. Parts of speech are also known as word classes or lexical categories.
6	Case folding	Case folding describes the process of consolidating multiple spellings of a single word that differ only in capitalization. This normalization technique is also known as case normalization.
7	Chunking	Chunking (or text chunking) is a type of shallow parsing that analyses a sentence by first identifying its constituent parts (nouns, verbs, adjectives, etc.) and then linking them to higher order units that have discrete grammatical meanings (noun groups or phrases, verb groups, etc.).
8	Cleaning	Text cleaning is the process of preparing raw text for NLP (Natural Language Processing) so that machines can understand human language.
9	Filtering	It is the process of removing stop words or any unnecessary data from the sentence.
10	Lemmatization	Use a vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the lemma.
11	Tokenization	The process of breaking a stream of text into words, phrases, symbols, or other meaningful tokens. Related terms: Word Segmentation
12	Semantic Role Labelling (SRL)	The process of detecting the semantic arguments linked to the predicate or verb of a sentence and their classification into their specific roles. Related Terms: Semantic parsing, semantic trees, shallow parsing, and shallow semantic analysis.
13	Stop-Word Removal	Words that are filtered out before or after processing of natural language data (text).

14	Stemming	A crude heuristic process that chops off the ends of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the lemma.
15	Sentence clustering	Sentence clustering is to automatically group textual documents (for example, documents in plain text, web pages, emails and etc) into clusters based on their content similarity.
16	Latent Semantic Analysis (LSA)	A mathematical practice that helps classify and retrieve information on particular key terms and concepts using singular value decomposition (SVD). Related Term: Latent Semantic Indexing (LSI)
17	Named Entity Recognition (NER)	Subtask of information extraction that is based on finding and classifying named entities in a certain text into predefined categories or classes such as the names of persons, organisations, locations, etc. Related terms: Entity Identification, Concept Extraction.
18	Parsing	Parsing is the process of analyzing a sentence's syntax and its underlying structure to extract meaning from it.
19	Punctuation removal	Removing punctuation marks.
20	Semantic Analysis	To identify and label semantically relevant components and relations in the text. This entails identifying the meaning of a certain word or phrase in a context and the relationship between words or terms.
21	Sentence Embedding	A generalized word2vec method, for representing documents as a vector. Related term: Doc2Vec
22	Sentiment Analysis	The process of computationally identifying and categorising opinions expressed in a piece of text.
23	Syntactic Analysis	To analyse the syntactic structure of a sentence to represent the relationship between its components. Different representation structures can be used, such as the parse tree, or the dependency parsing graph.
24	Syntactic dependency	A syntactic dependency is a relation between two words in a sentence, with one word being the governor and the other being the dependent of the relation. Syntactic dependencies often form a tree.

25	Term Frequency-Inverse Document Frequency (TF-IDF)	A statistical measure that evaluates how relevant a word is to a document in a collection of documents.
26	Text mining	Text mining is the process of deriving high-quality information from text.
27	Word Embedding	One of the most popular techniques to learn word embeddings using shallow neural networks. Word embeddings are vector representations of a particular word. Related terms: Word2Vec

4.3 Algorithms and Models

In Figure 4.4, the most frequently used NLP algorithm is Singular Vector Decomposition (used by 14 studies), followed by Naive Bayes (used by 14 studies), Term Frequency-Inverse Document Frequency (TF-IDF) (used by 13 studies), Word2Vec (used by 9 studies), Bag of Words (used by 6 studies), n-gram (used by 5 studies) and Random Forest (used by 4 studies).

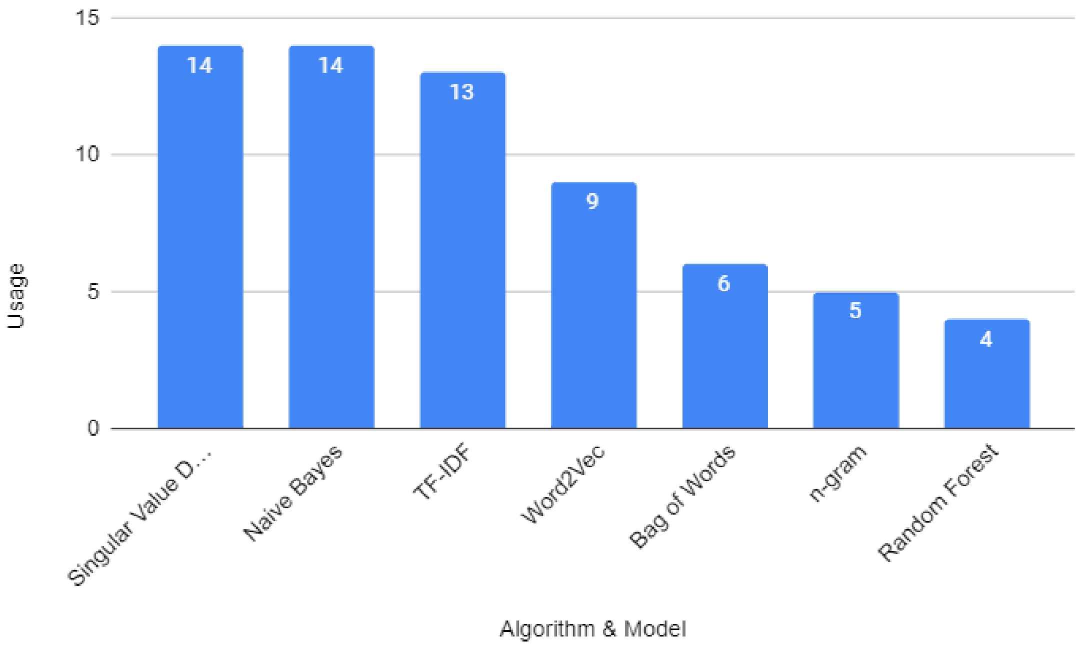

Figure 4.4: Most frequently used NLP algorithms and models

Table 4.3 shows the list of algorithms and models identified in our work. There are 24 different algorithms and models described in Table 4.3.

Table 4.3: List of algorithms and models

ID	**Algorithms and Models**
1	Bag of Words
2	n-gram
3	Support Vector Machines (SVM)
4	Weighted Finite-State Transducers (WFST)
5	Bag of Frames
6	BERT
7	Continuous Bag of Words
8	Continuous Skip gram
9	Fisher Kernel
10	Term Frequency-Inverse Document Frequency (TF-IDF)
11	Gaussian Naive Bayes
12	GPT
13	Hierarchical Agglomerative clustering algorithm (HAC)
14	K-nearest neighbour (KNN)
15	Logistic Regression (LR)
16	Multinomial Naive Bayes
17	Random Forest
18	Singular Value Decomposition
19	String search Algorithm
20	Skip gram
21	Tree traversal
22	Word2Vec
23	Multi-Pass sieve text classification
24	Vector Space Model (VSM)

4.4 Libraries and Tools

Figure 4.5 shows that the most frequently used NLP tool is NLTK (used by 15 studies), while the next most used tools are Spacy (used by 7 studies), Stanford CoreNLP (used by 7 studies), WordNet (used by 5 studies), Scikit-Learn (used by 4 studies), and Gensim (used by 3 studies).

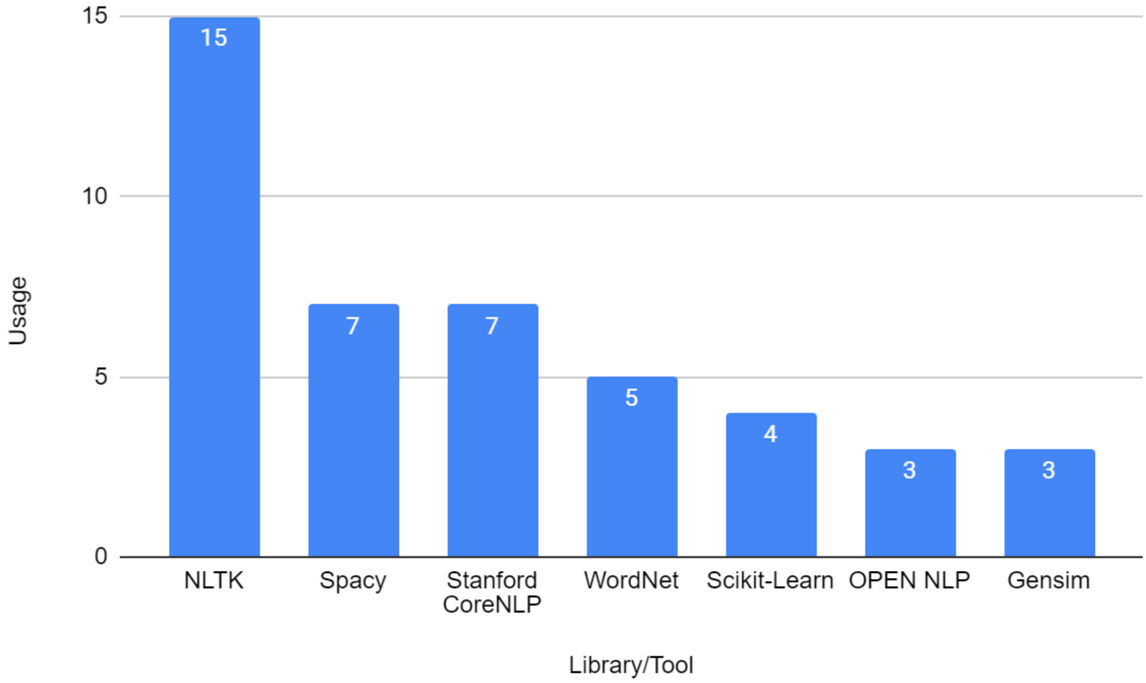

Figure 4.5: Most frequently used NLP libraries and tools

Table 4.4 shows the list of libraries and tools used in our work.

Table 4.4: List of libraries and tools

ID	Library/Tool
1	Apache PDFBox
2	GATE
3	Gensim
4	NLTK
5	OpenNLP
6	Pandas
7	PyLDAvis
8	Scikit-learn
9	Regex
10	Sentiwordnet
11	Spacy
12	Stanford CoreNLP
13	Textblob
14	WordNet
15	Quars

4.5 Countries of the Authors

Figure 4.6 shows the data on authors' contributions in our 57 articles published around the world. The figure shows that around 67 authors from 25 countries published 57 articles, with Italy leading the chart with 8 authors, followed by the US and China with 7 authors, Germany with 6 authors, Canada with 4 authors, and India with 4 authors.

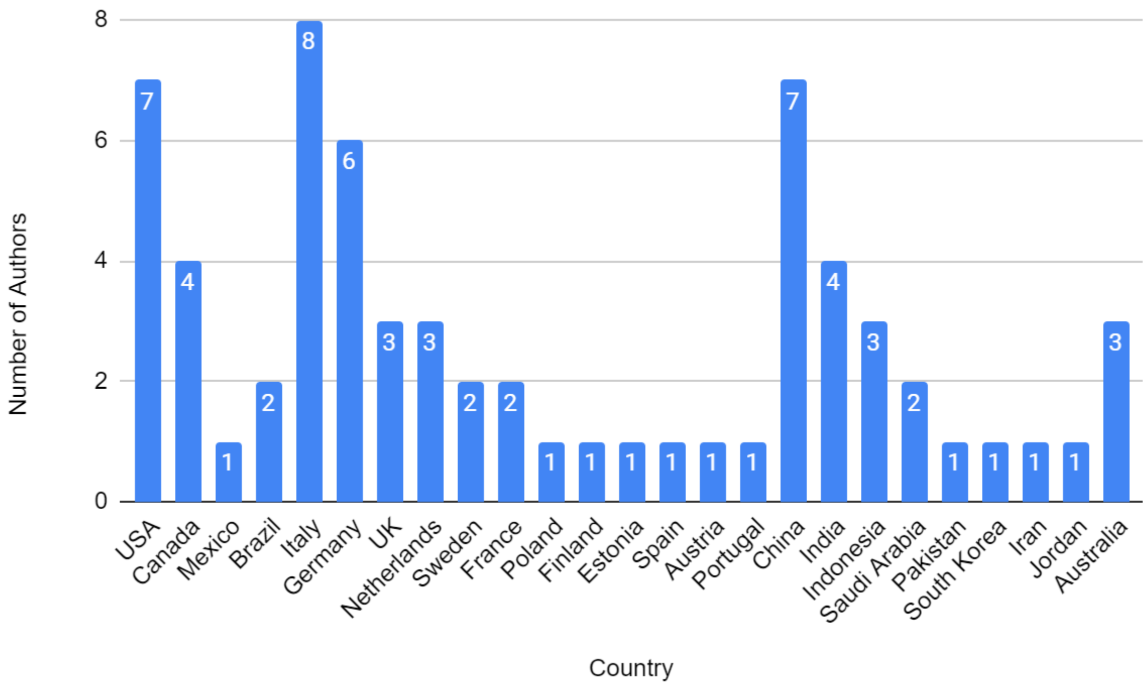

Figure 4.6: Articles published by authors around the world

5 Limitations

Some of the major limitations regarding our study are listed below.

- Time restrictions: There is a predefined time limit for completing the non-technical project in our course curriculum. Due to that, we need to speed up the process a little bit with the data and sources that were available within the given time limits.
- Limited knowledge about existing NLP methods: There are a number of different techniques, algorithms, methods, etc., and many of them were completely new for us, so identifying all of them was a difficult task as all the different techniques, algorithms, methods, etc. were identified manually from all the documents.
- Source restrictions: After trying different data sources with different filtering options, the number of results was huge, so applying strict filters such as year limitations from 2018 - 2023 finally allowed us to get a good number of results.
- Keyword restrictions: Keywords play a vital role in a literature review, as stricter keywords can give less results, while weaker keywords can give huge and irrelevant results. So in our survey, we used limited and stricter keywords to get a sufficient amount of good results.
- Limitation of human capacity: This review was carried out by one single person with the help of one supervisor, so collecting, filtering, and processing the complete data is a bit of a difficult task in a given time limit, but with the help of other people, maybe this literature review can be further extended to big data and more accurate results could be obtained.

6 Summary and Conclusion

In this paper, we perform a literature review on requirements extraction using natural language processing. The Scopus database and Zhao's references was used to collect different articles by formulating a search expression, the study was performed on articles published between January 2018 to May 2023.The target of this literature review was to find out the current technologies used in the field of requirements extraction using natural language processing (NLP). This review presented 92 articles related to software requirements extraction, out of which 57 studies were reviewed thoroughly.

Different NLP techniques, such as part-of-speech tagging, named entity recognition, semantic analysis, and syntactic parsing, as well as NLP algorithms, NLP language libraries, etc., were also reviewed from 57 textual documents. Also, different algorithms and models, programming languages, and NLP libraries/tools were analysed.

After a detailed analysis of our work, it is clear that there is a growing number of literature in this field that showcases the growing research interest in this field. Among the identified NLP techniques are a lot of basic NLP methods using statistical approaches, only a few language models, and more recent and sophisticated approaches. The NLP library from Stanford is no longer the most commonly used library; instead, NLTK is very strong, and Spacy has been strongly emerging during the last few years (compared to Zhao's paper). We have identified that the most preferred programming languages are Python and Java, as expected in the ML domain. The identified countries show that ML topics are strong in the US and China, but also in Europe, such as Italy and Germany, and will keep on growing around the world in the near future. Overall, there seems to be a lot of potential for the application of recent NLP techniques to requirement extraction tasks in the coming years.

References for Literature Review

[1] Liping Zhao, Waad Alhoshan, Alessio Ferrari, Keletso J Letsholo, Muideen A Ajagbe, Erol-Valeriu Chioasca, and Riza T Batista-Navarro. Natural language processing for requirements engineering: A systematic mapping study. *ACM Computing Surveys (CSUR)*, 54(3):1–41, 2021.

[2] Lenis R Wong, David S Mauricio, Glen D Rodriguez, et al. A systematic literature review about software requirements elicitation. *Journal of Engineering Science and Technology*, 12(2):296–317, 2017.

[3] Liping Zhao, Waad Alhoshan, Alessio Ferrari, and Keletso J Letsholo. Classification of natural language processing techniques for requirements engineering. *arXiv preprint arXiv:2204.04282*, 2022.

[4] Nishank Nitin Navelkar. Literature survey on fuel cell modeling: Automotive applications. Technical report, Otto von Guericke Universität, Magdeburg, de, 10 2022.

References for Selected Studies

[S1] Z. Shakeri Hossein Abad, V. Gervasi, D. Zowghi, and K. Barker. Elica: An automated tool for dynamic extraction of requirements relevant information. In *2018 5th International Workshop on Artificial Intelligence for Requirements Engineering (AIRE)*, pages 8–14, 2018.

[S2] Z.S.H. Abad, M. Rahman, A. Cheema, V. Gervasi, D. Zowghi, and K. Barker. Dynamic visual analytics for elicitation meetings with elica. In Günther Ruhe, Walid Maalej, and Daniel Amyot, editors, *2018 IEEE 26th International Requirements Engineering Conference*, Piscataway, NJ, 2018. IEEE.

[S3] Zahra Shakeri Hossein Abad, Vincenzo Gervasi, D. Zowghi, and B. H. Far. Supporting analysts by dynamic extraction and classification of requirements-related knowledge. In *2019 IEEE/ACM 41st International Conference on Software Engineering: companion proceedings*, volume 2019-May, Piscataway, NJ, 2019. IEEE.

[S4] H. Akay, M. Yang, and S.-G. Kim. Automating design requirement extraction from text with deep learning. In *Proceedings of the ASME 2021 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC-CIE2021)*, volume 3B-2021, New York, NY, USA and Red Hook, NY, 2021. the American Society of Mechanical Engineers and Curran Associates Inc.

[S5] J. Allen and S. Reddivari. Efficient parallel wikipedia internal link extraction for nlp-assisted requirements understanding. In Hong Va Leong, editor, *2022 IEEE 46th Annual Computers, Software, and Applications Conference*, Piscataway, NJ, 2022. IEEE.

[S6] N. Alturaief, H. Aljamaan, and M. Baslyman. Aware: Aspect-based sentiment analysis dataset of apps reviews for requirements elicitation. In *2021 36th IEEE/ACM International Conference on Automated Software Engineering workshops*, Piscataway, NJ, 2021. IEEE.

[S7] A. Alwadain and M. Alshargi. Crowd-generated data mining for continuous requirements elicitation. *INTERNATIONAL JOURNAL OF ADVANCED COMPUTER SCIENCE AND APPLICATIONS*, 10(9):45–50, 2019.

[S8] A. E. Amalia and M. Z. Naf'An. Mining user reviews for software requirements of a new mobile banking application. In *Machine learning for data science*, Piscataway, NJ, 2021. IEEE.

[S9] D. Arruda, M. Marinho, E. Souza, and F. Wanderley. A chatbot for goal-oriented requirements modeling. In Sanjay Misra, editor, *Computational Science and Its Applications - ICCSA 2019*, volume 11622 LNCS of *Lecture Notes in Computer Science Ser*, pages 506–519, Cham, 2019. SPRINGER INTERNATIONAL PUBLISHING AG.

[S10] M. Asif, I. Ali, M. S. A. Malik, M. H. Chaudary, S. Tayyaba, and M. T. Mahmood. Annotation of software requirements specification (srs), extractions of nonfunctional requirements, and measurement of their tradeoff. *IEEE Access*, 7:36164–36176, 2019.

[S11] D. Bajaj, A. Goel, S. C. Gupta, and H. Batra. Muce: a multilingual use case model extractor using gpt-3. *International Journal of Information Technology (Singapore)*, 14(3):1543–1554, 2022.

[S12] A. M. Ebrahimi and A. A. Barforoush. Preprocessing role in analyzing tweets towards requirement engineering. In *27th Iranian Conference on Electrical Engineering*, Piscataway, NJ, 2019. IEEE.

[S13] Alessandro Fantechi, Alessio Ferrari, Stefania Gnesi, and Laura Semini. Hacking an ambiguity detection tool to extract variation points: an experience report. In *Proceedings of the 12th International Workshop on Variability Modelling of Software-Intensive Systems*, pages 43–50, 3168381, 2018. ACM.

[S14] A. Fantechi, A. Ferrari, S. Gnesi, and L. Semini. Requirement engineering of software product lines: Extracting variability using nlp. In *2018 IEEE 26th International Requirements Engineering Conference (RE)*, pages 418–423, 2018.

[S15] A. Ferrari, A. Esuli, and S. Gnesi. Identification of cross-domain ambiguity with language models. In Eduard C. Groen, Rachel Harrison, Pradeep K. Murukannaiah, and Andreas Vogelsang, editors, *2018 5th International Workshop on Artificial Intelligence for Requirements Engineering*, Piscataway, NJ, 2018. IEEE.

[S16] A. Ferrari and A. Esuli. An nlp approach for cross-domain ambiguity detection in requirements engineering. *Automated Software Engineering*, 2019.

[S17] E. Friesen, F. S. Bäumer, and M. Geierhos. Cordula: Software requirements extraction utilizing chatbot as communication interface. In *NPL4RE: 1st Workshop on Natural Language Processing for Requirements Engineering*, volume 2075, 2018.

[S18] T. Gemkow, M. Conzelmann, K. Hartig, and A. Vogelsang. Automatic glossary term extraction from large-scale requirements specifications. In *2018 IEEE 26th International Requirements Engineering Conference (RE)*, pages 412–417, 2018.

[S19] I. Gräßler, D. Preus, L. Brandt, and M. Mohr. Efficient extraction of technical requirements applying data augmentation. In *2022 IEEE International Symposium on Systems Engineering (ISSE)*. IEEE, 10/24/2022 - 10/26/2022.

[S20] K. J. Gülle, N. Ford, P. Ebel, F. Brokhausen, and A. Vogelsang. Topic modeling on user stories using word mover's distance. In *7th International Workshop on Artificial Intelligence and Requirements Engineering*, Piscataway, NJ, 2020. IEEE.

[S21] H. Guo, Ö. Kafali, and M. Singh. Extraction of natural language requirements from breach reports using event inference. In *2018 5th International Workshop on Artificial Intelligence for Requirements Engineering (AIRE)*, pages 22–28, 2018.

[S22] I. Hajri, A. Goknil, L. C. Briand, and T. Stephany. Configuring use case models in product families. *Software and Systems Modeling*, 17(3):939–971, 2018.

[S23] A. Henriksson and J. Zdravkovic. A data-driven framework for automated requirements elicitation from heterogeneous digital sources. In Jānis Grabis, editor, *The Practice of Enterprise Modeling*, volume 400 of *Lecture Notes in Business Information Processing Ser*, pages 351–365, Cham, 2020. SPRINGER INTERNATIONAL PUBLISHING AG.

[S24] A. Henriksson and J. Zdravkovic. Holistic data-driven requirements elicitation in the big data era. *Software and Systems Modeling*, 21(4):1389–1410, 2022.

[S25] M. Jahan, Z.S.H. Abad, and B. Far. Detecting use case scenarios in requirements artifacts: A deep learning approach. In Hamido Fujita, Philippe Fournier-Viger, Moonis Ali, and Yinglin Wang, editors, *Advances and trends in artificial intelligence*, volume 13343 LNAI of *Lecture notes in computer science Lecture notes in artificial intelligence*, pages 682–694, Cham, 2022. Springer.

[S26] V. Jain, R. Malhotra, S. Jain, and N. Tanwar. Cross-domain ambiguity detection using linear transformation of word embedding spaces. In *NPL4RE: 3rd Workshop on Natural Language Processing for Requirements Engineering*, March 24, 2020.

[S27] N. Jha and A. Mahmoud. Using frame semantics for classifying and summarizing application store reviews. *Empirical Software Engineering*, 23(6):3734–3767, 2018.

[S28] J. A. Khan, L. Liu, and L. Wen. Requirements knowledge acquisition from online user forums. *IET SOFTWARE*, 14(3):242–253, 2020.

[S29] T. Kochbati, S. Gérard, S. Li, and C. Mraidha. From word embeddings to text similarities for improved semantic clustering of functional requirements. In *The 33rd International Conference on Software Engineering & Knowledge Engineering*, 2021.

[S30] J. Kuchta and P. Padhiyar. Extracting concepts from the software requirements specification using natural language processing. In *2018 11th International Conference on Human System Interaction (HSI)*, pages 443–448, 2018.

[S31] M. Lafi, B. Hawashin, and S. AlZu'bi. Eliciting requirements from stakeholders' responses using natural language processing. *CMES - Computer Modeling in Engineering and Sciences*, 127(1):99–116, 2021.

[S32] Tong Li, Fan Zhang, and Dan Wang. Automatic user preferences elicitation: A data-driven approach. In Erik Kamsties, Jennifer Horkoff, and Fabiano Dalpiaz, editors, *Requirements Engineering: Foundation for Software Quality*, Lecture Notes in Computer Science, pages 324–331. Springer International Publishing, 2018.

[S33] Yang Li. Feature and variability extraction from natural language software requirements specifications. In *Proceedings of the 22nd International Systems and Software Product Line Conference - Volume 2*, pages 72–78, 3236427, 2018. ACM.

[S34] C. Li, L. Huang, J. Ge, B. Luo, and V. Ng. Automatically classifying user requests in crowdsourcing requirements engineering. *Journal of Systems and Software*, 138:108–123, 2018.

[S35] A. Martinez, M. Melin, G. Koutsopoulos, and J. Zdravkovic. An empirical study on data-driven requirements elicitation: Reflections from nordic enterprises. In *The Practice of Enterprise Modeling 2021 Forum*, volume 3045, 2021.

[S36] F. Mokammel, E. Coatanéa, J. Coatanéa, V. Nenchev, E. Blanco, and M. Pietola. Automatic requirements extraction, analysis, and graph representation using an approach derived from computational linguistics. *SYSTEMS ENGINEERING*, 21(6):555–575, 2018.

[S37] I. Morales-Ramirez, F. M. Kifetew, and A. Perini. Speech-acts based analysis for requirements discovery from online discussions. *Information Systems*, 86:94–112, 2019.

[S38] P. V. Nistala, A. Rajbhoj, V. Kulkarni, S. Soni, K. V. Nori, and R. Reddy. Towards digitalization of requirements: generating context-sensitive user stories from diverse specifications. *Automated Software Engineering*, 29(1), 2022.

[S39] Z. Peng, P. Rathod, N. Niu, T. Bhowmik, H. Liu, L. Shi, and Z. Jin. Environment-driven abstraction identification for requirements-based testing. In Ana Moreira, Kurt

Schneider, Michael Vierhauser, and Jane Huang, editors, *29th IEEE International Requirements Engineering Conference*, Piscataway, NJ, 2021. IEEE.

[S40] Z. Peng, P. Rathod, N. Niu, T. Bhowmik, H. Liu, L. Shi, and Z. Jin. Testing software's changing features with environment-driven abstraction identification. *Requirements Engineering*, 27(4):405–427, 2022.

[S41] R.L.Q. Portugal, T. Li, L. Silva, E. Almentero, and J.C.S. Do Prado Leite. Nfrfinder: A knowledge based strategy for mining non-functional requirements. In *SBES '18: Proceedings of the XXXII Brazilian Symposium on Software Engineering*, September 17 - 21, 2018.

[S42] F. Qin, H. Xu, W. Zhang, L. Yuan, M. Li, Y. Liu, and Y. Chen. Voice of the customer oriented new product synthesis over knowledge graphs. In *Proceedings of the ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference - 2018*, volume 1A-2018, New York, N.Y., 2018. the American Society of Mechanical Engineers.

[S43] A. Rabinia, M. Dragoni, and S. Ghanavati. Towards integrating the flg framework with the nlp combinatory framework. In *PAL: Privacy-Enhancing Artificial Intelligence and Language Technologies*, volume 2335, 2019.

[S44] I. K. Raharjana, D. Siahaan, and C. Fatichah. User story extraction from online news for software requirements elicitation: A conceptual model. In *Knowledge evolution towards singularity of man-machine intelligence*, Piscataway, NJ, 2019. IEEE.

[S45] K. Rajbabu, Harshavardhan Srinivas, and S. Sudha. Industrial information extraction through multi-phase classification using ontology for unstructured documents. *Computers in Industry*, 100:137–147, 2018.

[S46] Iris Reinhartz-Berger and Mark Kemelman. Extracting core requirements for software product lines. *Requirements Engineering*, 2019.

[S47] F. Sarro, M. Harman, Y. Jia, and Y. Zhang. Customer rating reactions can be predicted purely using app features. In Günther Ruhe, Walid Maalej, and Daniel Amyot, editors, *2018 IEEE 26th International Requirements Engineering Conference*, Piscataway, NJ, 2018. IEEE.

[S48] Faiz Ali Shah, Kairit Sirts, and Dietmar Pfahl. Is the safe approach too simple for app feature extraction? a replication study. In Eric Knauss and Michael Goedicke, editors, *Requirements Engineering: Foundation for Software Quality*, Lecture Notes in Computer Science, pages 21–36. Springer International Publishing, 2019.

[S49] D. Siahaan, I. K. Raharjana, and C. Fatichah. User story extraction from natural language for requirements elicitation: Identify software-related information from online news. *Information and Software Technology*, 158, 2023.

[S50] T. Spijkman, B. Winter, S. Bansidhar, and S. Brinkkemper. Concept extraction in requirements elicitation sessions: Prototype and experimentation. In *NLP4RE: 4th Workshop on Natural Language Processing for Requirements Engineering*, volume 2857, 2021.

[S51] T. Spijkman, F. Dalpiaz, and S. Brinkkemper. Back to the roots: Linking user stories to requirements elicitation conversations. In Eric Knauss, editor, *30th IEEE International Requirements Engineering Conference*, volume 2022-August, Piscataway, NJ, 2022. IEEE.

[S52] T. Spijkman, X. de Bondt, F. Dalpiaz, and S. Brinkkemper. Summarization of elicitation conversations to locate requirements-relevant information. In Alessio Ferrari and Birgit Penzenstadler, editors, *Requirements Engineering: Foundation for Software Quality*, volume 13975 LNCS of *Lecture Notes in Computer Science*, pages 122–139, Cham, 2023. Springer Nature Switzerland and Imprint Springer.

[S53] Anjali Sree-Kumar, Elena Planas, Robert Claris, and #243. Extracting software product line feature models from natural language specifications. In *Proceedings of the 22nd International Systems and Software Product Line Conference - Volume 1*, pages 43–53, 3233033, 2018. ACM.

[S54] S. Wein and P. Briggs. A fully automated approach to requirement extraction from design documents. In *2021 IEEE Aerospace Conference (50100)*, volume 2021-March, Piscataway, NJ, 2021. IEEE.

[S55] K. Winter and S. Rinderle-Ma. Detecting constraints and their relations from regulatory documents using nlp techniques. In Hervé Panetto, editor, *On the Move to Meaningful Internet Systems. OTM 2018 Conferences*, volume 11229 LNCS of *Lecture Notes in Computer Science Ser*, pages 261–278, Cham, 2018. SPRINGER INTERNATIONAL PUBLISHING AG.

[S56] C. Wu, Z. H. Huang, Y. T. Yang, and Y. Liu. Requirement consistency and integrity verification method based on natural language processing. *Journal of Physics: Conference Series*, 1756(1), 2021.

[S57] A. Zaki-Ismail, M. Osama, M. Abdelrazek, J. Grundy, and A. Ibrahim. Rcm-extractor: an automated nlp-based approach for extracting a semi formal representation model from natural language requirements. *Automated Software Engineering*, 29(1), 2022.