Extraction of Useful Information from Unstructured Data in Software Engineering: A Systematic Mapping

Patrick R. Silva[1], Vinicius dos Santos[2], Érica F. Souza[1], Giovani V. Meinerz[1], Katia R. Felizardo[1], and Nandamudi L. Vijaykumar[3]

[1] Federal University of Technology - Paraná (UTFPR), Cornélio Procópio - Brazil
patrick.1985@alunos.utfpr.edu.br, {ericasouza, giovanimeinerz, katiascannavino}@utfpr.edu.br
[2] Universidade de São Paulo, São Carlos, São Paulo - Brazil
vinicius.dos.santos@usp.br
[3] National Institute for Space Research (INPE) and Federal University of São Paulo (Unifesp)
São José dos Campos, São Paulo - Brazil
vijay.nl@inpe.br,vijaykumar@unifesp.br

Abstract. Context: A large number of information is generated and manipulated in Software Engineering (SE) projects. The technology surrounding this domain is constantly evolving. To keep up with such evolution, developers share their knowledge and seek help from other developers by means of interactive and collaborative environments. Understanding and extracting knowledge from these environments can enable developers to identify useful information for the project. **Objective:** This work aims to identify the main textual analysis approaches to extract useful information in the SE. **Method:** To achieve the proposed objective, we conducted a Systematic Mapping (SM). **Results:** We analyzed 69 relevant primary studies addressing approaches to extract useful information in the SE. **Conclusion:** Among the main conclusions of this study, we can infer that discussion forums attracted a significantly attention in SE context and it becomes one of the main textual databases investigated to extract useful information.

Keywords: Software Engineering · Textual Analysis · Useful Information · Systematic Mapping

1 Introduction

One of the characteristics of Software Engineering (SE) projects is the large quantity of information that is generated and manipulated [22, 29]. Those involved in the project face problems such as: difficulty in systematizing the information generated throughout the software processes; difficulty in reusing the knowledge generated from one project to another; loss of intellectual capital of the organization; and the non-representation of knowledge [24]. In addition, SE

is made up of a huge number of constantly evolving tools, frameworks, and APIs. Technologies that are innovative, in just over a year, may become obsolete. So, to keep up with this ever-changing pace, developers share their knowledge and seek help from other developers in areas where they have less knowledge [28].

According to Abidi et al. (2009) [1], the social web paradigm can be useful for sharing knowledge through interactive and collaborative technologies. Questions & Answers (Q&A) like Stack Overflow, control tools like Jira, Redmine and Mantis, and source code sharing platforms like GitHub, are very rich sources of information. Through these environments, the entire software development process could be discussed and documented by team members, and most are texts written in natural language. Understanding and extracting knowledge from these environments can enable developers to identify useful information such as usage trends, previously created function facilities, best practices, lessons learned and new coding methodologies [17, 28].

Manual access to the large volume of textual data generated to search for useful information becomes an arduous and complex task. Identifying relevant knowledge for reuse is difficult and requires specific approaches to handling a large volume of information [1]. For this purpose, there is a huge range of Natural Language Processing (NLP) and Text Mining tools and techniques that can be used to assist in knowledge extraction. NLP and Text Mining are both interested in processing and extracting information from texts. NLP uses different levels of linguistic analysis responsible for understanding meaning and structure of a given text [10]. On the other hand, text mining extracts hidden information inside text data through pattern recognition [6].

The demands for reduced development time and increased reliability of software need for new approaches that provide support for development and decision making process. Knowledge extraction from text data set previously collected, for example, can bring a lot of useful information related to software development. Inserted in this context, this work has an objective to identify in the literature the main textual analysis approaches to extract useful information in SE domain, such as SE activities the study focuses on (e.g. Software Requirements, Software Testing, Development), textual database considered (e.g. Forums, Wiki, Tweets), techniques, algorithms and tools that were considered. In order to achieve the proposed objective, we conducted a Systematic Mapping (SM). An SM provides a broad overview of an area of research, to determine whether there is research evidence on a particular topic [12].

We believe that results from this SM can strongly help to identify a body of knowledge to support future research, learning as much as possible from other domains related to the topic, and providing a basis for other researchers as well as students who consider learning about and contributing to this area.

The remainder of this paper is structured as follows. Section 2 presents the related publications. Section 3 introduces the method used to conduct the research. Section 4 shows the main results from studies extraction and synthesis. Section 5 reports a general discussion to highlight some research points, their

implications, and limitations. Lastly, conclusions and future directions for this research are presented Section 6.

2 Related Work

We adopted concepts of a tertiary study to look for secondary studies that already investigated approaches to extract useful information in the SE considering unstructured data. Tertiary studies are reviews about other secondary studies [12]. In this study, we used the following search string: *("text mining" OR "text processing" OR "text classification" OR "knowledge extraction" OR "information extraction" OR "text cluster" OR "text data mining" OR "text association" OR "Natural Language Processing" OR "NLP") AND ("software engineering" OR "software organizations" OR "software development") AND ("systematic review" OR "literature review" OR "systematic mapping" OR "mapping study" OR "systematic map" OR "literature analysis").*

The search string was applied in the Scopus electronic database and 52 studies were returned. We analyzed each of the 52 studies and we found four studies that conducted a review on the same topic as our SM. The studies are briefly described as follows.

In Maqbool et al. [14], a Systematic Literature Review (SLR) was performed to investigate modern techniques, tools and trends for the generation of Business Process Modeling Languages (BPML) models from textual requirements by utilizing NLP techniques. Ahmad et al. [2] conducted a literature review to show how academics/practitioners can benefit from the valuable user-generated content in software repositories including the Q&A software developer community.

SLR was conducted in Ahsan et al. (2017) [3] to investigate the application of NLP techniques to generate test cases from requirements document. The main NLP techniques and tools used to extract textual information from requirements document were summarized in this study. Finally, in Shah and Pfahl [23], a mapping study was conducted to identify relevant information from the scientific literature about data sources, research contributions, and the usage of text analysis techniques for the improvement and evaluation of software quality. The authors summarized, in this study, the data sources which have been used, research contributions that have been made, text analysis techniques which have been employed and how does text analysis help in improving and evaluating software quality.

Each presented study conducted some kind of secondary study on extracting useful information in SE, but with different purposes. The main difference from the secondary studies identified with ours is that the studies found have specific scopes in SE, such as software testing, software quality or modeling languages. In our mapping study, the scope is much broader considering the various activities in SE. This gives a broader overview of where efforts for SE textual analysis research have been concentrated.

3 Systematic Mapping

SM has been used to provide a broad overview of the state of all relevant research available for a particular topic of interest. The research method in this study was defined based on Kitchenham and Charters [12]. The method involves three main phases. Firstly, a planning is elaborated. This planning refers to the pre-review activities, and establishes a protocol, defining the research questions, inclusion and exclusion criteria, sources of studies, search string, and mapping procedures. Then, in the conduction phase, search and selection activities are carried out in order to extract and synthesize data from studies included. Finally, in the final phase the results are written and disseminated to interested parties. Following, the main parts of the mapping protocol used in this work are presented.

Research Questions. This mapping study aims to answer the following Research Questions (RQs):

- RQ1. When and where were the studies published?
- RQ2. Which software engineering activities are the studies concentrated on?
- RQ3. What is the textual database considered for the study?
- RQ4. What techniques, algorithms, methods and tools were considered?
- RQ5. What are the main challenges/difficulties reported in the studies?

Inclusion and Exclusion Criteria. The selection criteria are organized in one Inclusion Criterion (IC) and eight Exclusion Criteria (EC). The inclusion criterion is: (IC1) The study must present textual analysis initiatives to extract useful information in the context of Software Engineering. The exclusion criteria are: (EC1) The study does not have a abstract; (EC2) The study is just an abstract, not having a full text; (EC3) The study is not a primary study, such as editorials, summaries of keynotes, workshops, and tutorials; (EC4) The study is not written in English; (EC5) The study is an older version (less updated) of another study already considered; (EC6) The full paper is not available; (EC7) The study must present textual analysis initiatives in the English language; and (EC8) Does not meet Inclusion Criterion (CI1).

Keywords and Search String. The search string considered two areas, Textual Analysis and Software Engineering (see Table 1), and it was applied in three metadata fields (namely, title, abstract and keywords).

Source. We chose to work with Scopus[4] database. This source is considered the largest abstract and citation database of peer-reviewed literature, with more than 60 million records. Scopus attaches papers of other international publishers, including Cambridge University Press, Institute of Electrical and Electronics Engineers (IEEE), Nature Publishing Group, Springer, Wiley-Blackwell, and Elsevier. The studies returned from Scopus database were cataloged and stored appropriately. This catalog helped us in the classification and analysis procedures.

[4] *https://www.scopus.com*

Table 1: Keywords for the search string

Areas	Keywords
Information Extraction	"text mining", "text processing", "text classification", "knowledge extraction", "information extraction", "text cluster", "text data mining", "text association", "Natural Language Processing", "NLP"
Software Engineering	"software engineering", "software organizations", "software development"
Search String: ("text mining" OR "text processing" OR "text classification" OR "knowledge extraction" OR "information extraction" OR "text cluster" OR "text data mining" OR "text association" OR "Natural Language Processing" OR "NLP") AND ("software engineering" OR "software organizations" OR "software development")	

Assessment. Before conducting the mapping study, we tested the mapping protocol in order to verify its feasibility and adequacy, based on a pre-selected set of studies considered relevant to our investigation. First, the review process was conducted by the first author of this paper, and, only then, the other authors carried out the review validation using different samples (approximately 30% each).

4 Data Extraction and Synthesis

The main steps performed in this mapping study are shown in Figure 1.

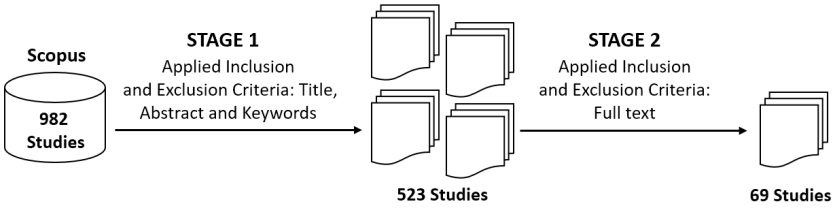

Fig. 1: Selection process

We considered the studies published in the last 10 years (2009 until April 2019). As a result, 982 publications were returned. The selection process was divided into two stages. In the 1^{st} stage, inclusion and exclusion criteria were applied considering the title, abstract and keywords, so 459 publications (approximately 46%) were eliminated. In the 2^{nd} stage, the exclusion criteria were applied considering the entire text, resulting in 69 studies (approximately 7% of total articles). As a result, we got a final set of 69 studies[5].

[5] Selected studies available at: *http://bit.ly/2NKq01s*

It is worth mentioning that in this SM the focus was on studies that used textual mining and NLP techniques to extract useful knowledge in order to be reused in software organizations. During the selection process, some primary studies returned, although using the same techniques for textual analysis, proposed processes to automatically generate UML diagrams from software requirements documents. The studies focus were solely in automation processes. However, that was not the scope of our focus. Studies with this objective were eliminated by the selection criteria.

We analyzed each of 69 studies in order to answer the RQs. The studies classification was made from the RQs answers. The same answers were grouped and consequently a category was defined. For some RQs in data extraction and synthesis the sum of all classifications might be greater than the total number of papers in the SM. This occurs because a given paper can answer a RQ with more than one target of our investigation. In addition, in some RQs, elements that were mentioned once were grouped into "Others" category. However, the full version of data extraction is available for consultation[6]. Next, we present the main findings from data extraction and synthesis.

RQ1. When and where have the studies been published?

A distribution of the 69 studies selected over the past ten years is shown in Figure 2. As Figure 2 demonstrates, although research on textual analysis initiatives intended to detect useful information is not recent, in SE domain the number of studies was moderate between 2010 and 2015, with a considerable increase between 2016 and 2018. Considering that the number of publications in the year 2019 is also considerable, our work, however, shows a smaller number as we consider only studies published until April 2019.

Analyzing the publication vehicle, conferences were the main communication channel, representing 72% (50 studies) of publications. The journals occupied 11% (8 studies), symposium 10% (7 studies) and workshop 6% (4 studies).

RQ2. Which software engineering activities are the studies concentrated on?

RQ2 relates to which activity of SE the study addressed. As shown in Figure 3, Software Development was the activity in which most of the studies were concentrated (54% - 37 studies). In Nembhard et al. (2018) [19], for example, text mining was used to extract knowledge from open source code in order to categorize and structure source code. By mining a subset (over 600.000 Java files) that contains over 70.000 open source projects, the authors presented that useful patterns can be extracted from source code and that these patterns can be used to create a recommends system to help programmers avoid unsafe practices.

[6] Data extraction available at: *http://bit.ly/2PiNQlC*

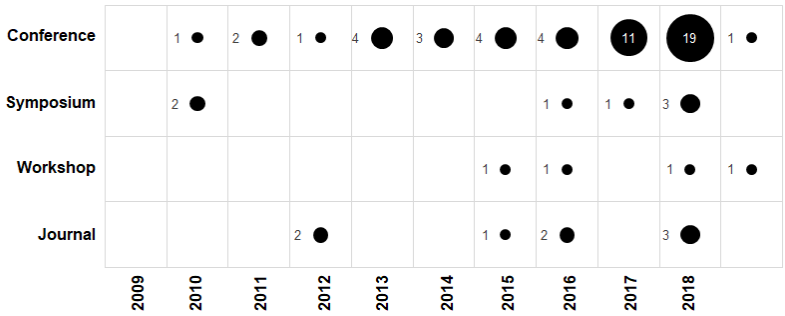

Fig. 2: Distribution of the selected studies over the years

Another example is the study conducted in Baquero et al. (2017) [5] in which, mechanisms for extracting implicit knowledge present in the Stack Overflow[7] questions were explored. In particular, the study motivation was to extract information about programming languages and their relationships. The proposed approach creates a classifier model that predicts the programming language using the content (text excerpts and source code) of a question. The method was evaluated on a set of 18.000 questions related to 18 different programming languages. The results showed that it is possible to extract non-evident information from a discussion forum, such as Stack Overflow.

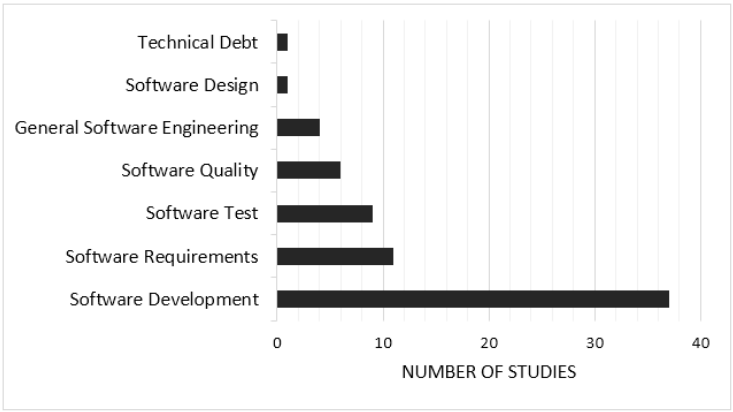

Fig. 3: Software Engineering activities analyzed

In relation to the other SE activities, software requirements occupied 16% (11 studies), software testing 13% (9 studies), software quality 9% (6 studies),

[7] *https://pt.stackoverflow.com*

general area of SE 6% (4 studies), technical debt (implied cost of rework) and software design both with 1% (1 study). Most studies proposed knowledge discovery to aid in some task involving decision making, for example, analysis of programmers comments in several project activities in order to detect behavior patterns and improve development, information reuse from old projects or identify which technology is most used [8, 9].

RQ3. What is the textual database considered for the study?

RQ3 focuses on identifying which textual database was considered in the selected studies. As shown in Figure 4, discussion forums are the most commonly used (33% - 23 studies), followed by requirements documents (14% - 10 studies), bug reports (8% - 6 studies), source code (7% - 5 studies), source code hosting platform (6% - 4 studies), code review comments (4 % - 3 studies), projects management tools (4% - 3 studies), manuals and tutorials (4% - 3).

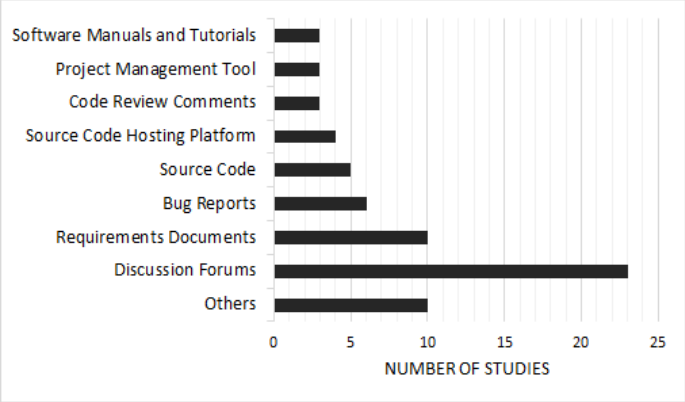

Fig. 4: Textual databases considered

Discussion forums have been the most used database for extracting useful information in SE. Out of the 23 studies that considered discussion forums, 19 used Stack Overflow as the basis for applying textual analysis techniques. Stack Overflow is a Q&A website for software development. According to Liu et al. (2018) [13], Stack Overflow's Q&A base has been increasingly used by professional software developers to get answers to their software development questions.

Stack Overflow was founded in 2008 and is the largest Q&A website for computer programming. This website currently contains a base of over 16 million questions and 24 million answers [15]. The large information data base maintained by Stack Overflow allows several searches for knowledge by researchers in SE. In the study conducted by Xu et al. (2017) [30], for example, the authors proposed a tool, called AnswerBot, that aims to help developers to quickly

capture key information from various relevant posts from a technical question before reading the details of posts.

In addition to the discussion forums, another textual source presented by the selected studies was the software requirements documents. In Verma et al. (2013) [27], for example, NLP techniques were used to extract useful knowledge from requirements documents in order to represent them as a simple graph-based structure. According to the authors, this representation helps in understanding and identifying useful information from textual requirements.

Further databases identified were grouped into category "Other" (14% - 10 studies), for example, Wikis [18] and Tweets [25].

RQ4. What techniques, algorithms and tools were considered?

For RQ4, we investigated which techniques, algorithms and tools were used to support textual analysis for information extraction. Figure 5 presents the main techniques used in the selected studies. The most commonly used techniques were those used in the text preprocessing (31% - 22 studies), for example, stop word removal, tokenization, Lemmatization and Stemming. We consider this set of techniques as a single category "Text Preprocessing". Other techniques often mentioned are those related to text vectorization (13% - 9 studies), for example, Word2vec, Bag of Words (BOW) and Word Embeddings. We also consider this set of techniques in a single category "Text Vectorization Techniques". Other techniques identified were Latent Dirichlet Allocation (LDA) (5% - 4 studies), Support Vector Machine (SVM) (4% - 3 studies), Named Entity Recognition (NER) (2% - 2 studies), Naive Bayes (2% - 2 studies) and Latent Semantic Indexing (LSI) (2% - 2 studies). In the category "Others" (18% - 13 studies), techniques such as Doc2vec, ELICA and exploratory data analysis (EDA) were grouped.

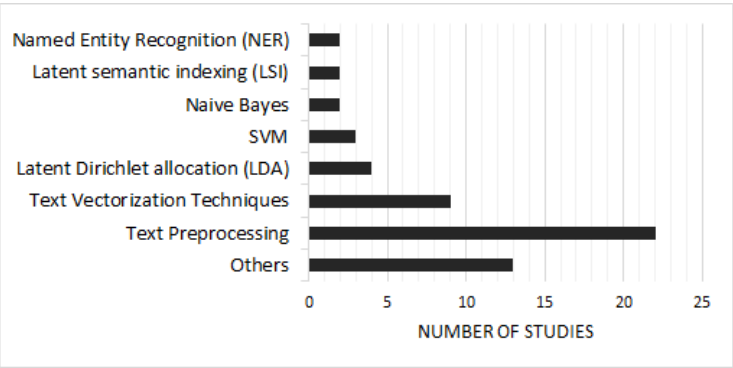

Fig. 5: Techniques used in textual analysis

Figure 6 presents the different algorithms used. The most commonly used algorithms was Term Frequency-Inverse Document Frequency (TF-IDF) with 72% (10 studies). TF-IDF algorithm maps the most relevant terms by means of their frequency, taking into consideration the frequency of this same term in a set of text analyzed [21].

We also map the tools mentioned in the selected studies. The most used tool was the Stanford CoreNLP[8] with 29% (13 studies). This tool provides a set of technologies to work with human language. For example, it gives the base forms of words, their parts of speech, normalize dates, times, and numeric quantities, mark up the structure of sentences in terms of phrases and syntactic dependencies, indicate which noun phrases refer to the same entities, indicate sentiment, extract particular or open-class relations between entity mentions.

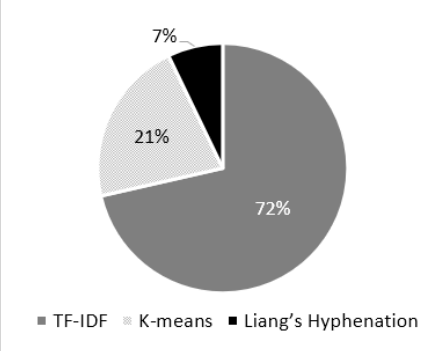

Fig. 6: Algorithms

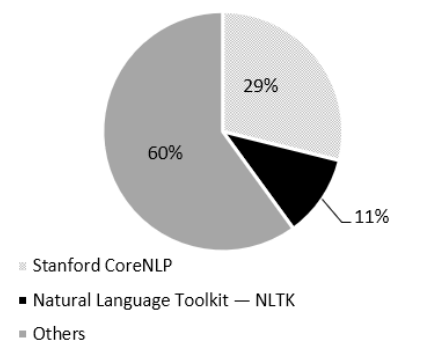

Fig. 7: Tools

Another widely used tool is the Natural Language Toolkit (NLTK)[9] (11% - 5 studies). NLTK is a tool for building Python programs to work with human language data. It provides interfaces to help to user work and a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries.

In "Others" category, tools such as Overlap, Gate, Scrapy, HDSKG and Sadge were mentioned.

RQ5. What are the main challenges/difficulties reported in the studies?

Although textual analysis can have several benefits in SE, some problems are also faced. The selected articles were analyzed for challenges, difficulties and problems, mentioned by the authors to identify useful information from textual

[8] *https://stanfordnlp.github.io/CoreNLP/*
[9] *https://www.nltk.org/*

databases, among the main challenges detected, we can highlight: (i) applying NLP techniques to software artifacts is complex since the artifacts have unique characteristics that not found in other natural language text [26]; (ii) software projects that are poorly documented [16]; and access to a set of industry scale domain documentations is a difficulty. Many organizations do not support the idea of publicly releasing the artifacts of their domain engineering practices. It becomes difficult to obtain real world requirement engineering documents for research purposes [4].

5 Discussion

Software engineering activities produce large amounts of unstructured data. Useful information can be extracted from such data to support SE activities, such as software development, software requirements or software test, as can be seen in RQ2. Software development activity, in particular, features many studies interested in exploring the knowledge that can be generated by considering aspects of source code, for example, language characteristics or comments in code [11].

The textual analysis in unstructured databases in SE, as presented in RQ3, is directly related to the SE activities, for example, the search for relevant information within discussion forums. Most studies that mention as purpose the use of this type of textual database are related to the software development activity. These studies look for behavioral patterns, especially in the comments from software developers. Online forums contain extensive valuable information that can aid in software development. Discussion forums are becoming rich repositories of valuable programming information that many experienced developers learn to revisit frequently [31].

Discussion forums are effective environments for facilitating the sharing of tacit knowledge among developers. Forums have become important repositories of knowledge for the following reasons: (i) useful knowledge can be generated and captured during discussions [7]; and (ii) it is considered a major challenge for knowledge management to convert tacit knowledge to explicit knowledge. Discussion forums can help in this knowledge conversion [20]. So, information extraction have been investigated to extract significant information from these unstructured data to aid in such development activities as the provision of method descriptions and the speed reading of bug reports [31].

Software developers extensively use Stack Overflow for knowledge sharing on software development. Thus, SE researchers have started mining the unstructured data present in certain software repositories including the Q&A software developer community Stack Overflow, to improve software development [2].

Extracting information for guidance of analyses is a real challenge. Those who want to follow the trends as well as predict them or even identify or detect patterns need to focus on the area of Artificial Intelligence. NLP and Text Mining, in particular, help in the forming, through unstructured data, patterns and associations useful for production of knowledge. Research conducted by EMC

Digital Universe[10], world leader of data storage, says that by 2020 there will be around 44 zettabytes (or 44 trillion GBs) of data stored. Therefore, Data Mining will become indispensable and new techniques, algorithms and tools for working with this amount of data to extract useful information are increasingly emerging.

In several areas, as in SE, researches are invested in creating techniques, algorithms or tools that are increasingly effective in extracting information from a textual database. In SE, as can be observed in RQ4, a large number of techniques can be used. With respect to tools, there are several proposals of new tools or an extension of existing ones, for example COLUA, a tool developed using python NLTK.

The research area in textual analysis has grown a lot in the last years in SE (RQ1). SE professionals have noted the benefits to the organization in identifying useful knowledge in what is textually generated by team members. In addition, the area enables researchers to focus their efforts on improving the various fronts involving this type of research, such as new techniques, tools or textual databases to be explored.

5.1 Threats to validity

Initially, our review was limited to Scopus database. Although Scopus is considered the largest abstract and citation database, it is possible that some valuable studies were left out from our analysis, since we considered papers indexed by just one database. Even so, we tried to overcome this limitation by considering papers from a control group to calibrate the search string. The categorization of papers was based on research questions. Thus, we believe that the studies discussed in this mapping provide an overview of empirical research on outcomes of existing research on textual analysis initiatives to extract useful information in unstructured databases in SE.

In addition, the study selection and data extraction were performed by the first author and some subjectivity could have been embedded. So, in order to reduce subjectivity, the other authors received different samples from the studies (about 30%) to conduct the same stages. The samples were compared to detect possible bias.

6 Conclusions

In this paper, we have reported the results of a systematic mapping on textual analysis initiatives to extract useful information in SE domain, such as the SE activities the study focuses, database, techniques, algorithms and tools that were considered.

The major contribution of this study was to summarize and highlight the main aspects associated with the search for knowledge in textual databases in

[10] https://www.emc.com/leadership/digital-universe/2014iview/executive-summary.htm

SE. The results showed in this research could be of interest to several researchers involved with the topics covered in this study. So, we believe that our summarization can help to direct researchers in their future research providing a pointer to appropriately position new activities in this research topic.

Future directions include conducting research on how the knowledge extracted from the selected studies databases has been represented/codified, for example, knowledge graph, cognitive mapping, decision trees, knowledge taxonomies, and task analysis. Knowledge representation serves the pivotal role of allowing what is collectively known to be shared and used. In addition, we intend to conduct research on the use of extracted information in SE databases and how it is represented in practice within SE organizations.

References

1. Abidi, S.S.R., Hussini, S., Sriraj, W., T.S., Finley, G.A.: Knowledge sharing for pediatric pain management via a web 2.0 framework. Health Technology and Informatics **150**, 287–291 (2009)
2. Ahmad, A., Feng, C., Ge, S., Yousif, A.: A survey on mining stack overflow: question and answering (qa) community. Data Technol. Appl. **52**, 190–247 (2018)
3. Ahsan, I., Butt, W.H., Ahmed, M.A., Anwar, M.W.: A comprehensive investigation of natural language processing techniques and tools to generate automated test cases. In: Intern. Conference on Internet of things, Data and Cloud Comp. (2017)
4. Bagheri, E., Ensan, F., Dragan, G.: Decision support for the software product line domain engineering lifecycle. Automated Software Engineering (2012)
5. Baquero, J., Camargo, J., Restrepo-Calle, F., Aponte, J., González, F.: Predicting the programming language: Extracting knowledge from stack overflow posts. In: Colombian Conference on Computing. pp. 199–210 (2017)
6. Cambria, E., White, B.: Jumping nlp curves: A review of natural language processing research. IEEE Computational Intelligence Magazine **9**, 48–57 (2014)
7. Falbo, R.A., Arantes, D.O., Natali, A.C.C.: Integrating knowledge management and groupware in a software development environment. In: International Conference on Practical Aspects of Knowledge Management. pp. 94–105. Austria (2004)
8. Flisar, J., Podgorelec, V.: Enhanced feature selection using word embeddings for self-admitted technical debt identification. In: 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA). pp. 230–233 (2018)
9. Huang, Y., Chen, C., Xing, Z., Lin, T., Liu, Y.: Tell them apart: distilling technology differences from crowd-scale comparison discussions. In: 33rd ACM/IEEE International Conference on Automated Software Engineering. pp. 214–224
10. Jurafsky, D., Martin, J.H.: Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition. Pearson Prentice Hall, Upper Saddle River, N.J. (2009)
11. Khatiwada, S., Kelly, M., Mahmoud, A.: Stac: A tool for static textual analysis of code. In: 24th International Conference on Program Comprehension (ICPC) (2016)
12. Kitchenham, B., Charters, S.: Guidelines for performing systematic literature reviews in software engineering. Tech. Rep. EBSE 2007-001, Keele University and Durham University, UK (2007)
13. Liu, M., Peng, X., Jiang, Q., Marcus, A., Yang, J., Zhao, W.: Searching stackoverflow questions with multi-faceted categorization. In: Asia-Pacific Symposium on Internetware (2018)

14. Maqbool, B., Azam, F., Anwar, M.W., Butt, W.H., Zeb, J., Zafar, I., Nazir, A.K., Umair, Z.: A comprehensive investigation of bpmn models generation from textual requirements - techniques, tools and trends. In: International Conference on Information Science and Applications (2018)
15. May, A., Wachs, J., Hannák, A.: Gender differences in participation and reward on stack overflow. Empirical Software Engineering (2019)
16. McBurney, P.W., Liu, C., McMillan, C.: Automated feature discovery via sentence selection and source code summarization. J. Softw. Evol. Process pp. 120–145 (2016)
17. Natali, A., Rocha, A., Travassos, G., Mian, P.: Integrating verification and validation techniques knowledge into software engineering environments. In: Jornadas Ibeoamericanas de Ingeniería del Software e Ingeniería del Conocimiento. pp. 419–430. Madri, Spain (2004)
18. Nawroth, C., Schmedding, M., Brocks, H., Kaufmann, M., Fuchs, M., Hemmje, M.: Towards cloud-based knowledge capturing based on natural language processing. In: Cloud Forward (2015)
19. Nembhard, F., Carvalho, M., Eskridge, T.: Extracting knowledge from open source projects to improve program security. SoutheastCon pp. 1–7 (2018)
20. Nonaka, I., Krogh, G.: Tacit knowledge and knowledge conversion: controversy and advancement in organizational knowledge creation theory. Organization Science **30**, 635–652 (2009)
21. Robertson, S.: Understanding inverse document frequency: On theoretical arguments for idf. Journal of Documentation - J DOC **60**, 503–520 (2004)
22. Rus, I., Lindvall, M.: Knowledge management in software engineering. IEEE Software **19**(3), 26–38 (2002)
23. Shah, F., Pfahl, D.: Evaluating and improving software quality using text analysis techniques - a mapping study. In: REFSQ Workshops (2016)
24. Souza, E.F., Falbo, R.A., Vijaykumar, N.L.: Knowledge management initiatives in software testing: A mapping study. Information and Software Technology **57**, 378–391 (2015)
25. Surapat, W., Prompoon, N.: Social clues powered, personalized software engineering messages classification. In: 10th International Symposium on Communications and Information Technologies. pp. 1114–1119 (2010)
26. Treude, C., Robillard, M.P., Dagenais, B.: Extracting development tasks to navigate software documentation. IEEE Transactions on Software Engineering **41**, 565–581 (2015)
27. Verma, R.P., Beg, M.R.: Representation of knowledge from software requirements expressed in natural language. In: International Conference on Emerging Trends in Engineering and Technology. pp. 154–158 (2013)
28. Vishal, J., Bansal, S.: Identifying trends in technologies and programming languages using topic modeling. In: 12th International Conference on Semantic Computing (ICSC). pp. 391–396 (2018)
29. Vliet, H.V.: Knowledge sharing in software development. In: 10th International Conference on Quality Software (2010)
30. Xu, B., Xing, Z., Xia, X., Lo, D.: Answerbot: automated generation of answer summary to developers' technical questions. In: 32Nd IEEE/ACM International Conference on Automated Software Engineering. pp. 706–716 (2017)
31. Zhang, Y., Hou, D.: Extracting problematic api features from forum discussions. In: 21st International Conference on Program Comprehension (ICPC). pp. 142–151 (2013)