Reverse Engineering Variability from Natural Language Documents: A Systematic Literature Review

Yang Li, Sandro Schulze, Gunter Saake
Otto-von-Guericke Universität Magdeburg, Germany
{yang.li,sandro.schulze,saake}@ovgu.de

ABSTRACT

Identifying features and their relations (i.e., variation points) is crucial in the process of migrating single software systems to software product lines (SPL). Various approaches have been proposed to perform feature extraction automatically from different artifacts, for instance, feature location in legacy code. Usually such approaches a) omit variability information and b) rely on artifacts that reside in advanced phases of the development process, thus, being only of limited usefulness in the context of SPLs. In contrast, feature and variability extraction from natural language (NL) documents is more favorable, because a mapping to several other artifacts is usually established from the very beginning. In this paper, we provide a multi-dimensional overview of approaches for feature and variability extraction from NL documents by means of a systematic literature review (SLR). We selected 25 primary studies and carefully evaluated them regarding different aspects such as techniques used, tool support, or accuracy of the results. In a nutshell, our key insights are that i) standard NLP techniques are commonly used, ii) post-processing often includes clustering & machine learning algorithms, iii) only in rare cases, the approaches support variability extraction, iv) tool support, apart from text pre-processing is often not available, and v) many approaches lack a comprehensive evaluation. Based on these observations, we derive future challenges, arguing that more effort need to be invested for making such approaches applicable in practice.

CCS CONCEPTS

•Computing methodologies →Natural language processing; •Software and its engineering →Software product lines;

KEYWORDS

Feature Identification, Variability Extraction, Reverse Engineering, Software Product Lines, Natural Language Documents, Systematic Literature Review

ACM Reference format:
Yang Li, Sandro Schulze, Gunter Saake. 2017. Reverse Engineering Variability from Natural Language Documents: A Systematic Literature Review. In *Proceedings of SPLC '17, Sevilla, Spain, September 25-29, 2017,* 10 pages.
DOI: 10.1145/3106195.3106207

978-1-4503-5221-5/17/09…$15.00
DOI: 10.1145/3106195.3106207

1 INTRODUCTION

Software product Line Engineering (SPLE) has been proposed as a large-scale development methodology that enables the efficient development of related software systems from a common set of core assets in a prescribed way [31, 32]. The resulting *software product line (SPL)* then comprises a set of software-intensive systems that can be distinguished by their commonalities and differences in terms of features. A feature in this context is a user-visible increment in functionality [4]. Based on (de-)selecting features, a particular variant can be tailored and generated based on the developed core assets.

Usually, software development does not start with an SPLE approach, because a) it induces high upfront costs (e.g., domain analysis) and b) it is mostly unclear whether a vast amount of variants is needed. Hence, traditional development approaches are preferred with ad-hoc mechanisms used to introduce variability. In particular, it is common practice in industry to apply *Clone&Own (CAO)*, that is, replicating an existing system and adapting it according to the new requirements [12, 35]. However, while this ad-hoc practice comes with low effort and is easy to use, information about commonalities and differences amongst the cloned systems is lost, thus, impeding the maintenance and evolution of the typically large number of variants. At some point, the aforementioned procedure becomes impractical, and thus, an SPLE approach is introduced, either using a *reactive* or *extractive* migration strategy [24]. A crucial point during this transition is to identify features and the variation points among them, i.e., how they relate to each other (e.g., alternative features or exclude/require relation between features). Especially for legacy systems that have been evolved over years, it is non-trivial to extract this information, and thus, automation is needed to support this step. Hence, *reverse engineering* techniques such as feature location and extraction are typically used to support the migration process.

While feature location has been subject to intensive research [11], their applicability for reverse engineering in an SPL context has two crucial limitations:

(1) Existing feature location techniques predominantly focus on single software systems, while information about variation points is missing. Hence, the extracted information is insufficient for migrating to a SPLE process.
(2) Since existing techniques focus solely on source code, additional effort may be required for feature extraction of other artifacts (e.g., requirements, models, documentation) due to missing traceability links from the source code.

We argue that these limitations can be cured by focusing on *software requirements specifications (SRS)* as primary artifact for feature and variability extraction. Due to considerable progress in *natural language processing (NLP)*, a variety of information, including variation points, can be extracted from such requirements, thus, addressing limitation 1. Moreover, requirements are the initial development artifact and usually traceability links to all other artifacts in later development phases, such as source code or test cases, are maintained. Hence, by extracting variability information from SRS documents, we can exploit these links for mapping of feature and variability information to other artifacts, thus, resolving limitation 2.

In this paper, we investigate the current state-of-the-art of feature and variability extraction from *natural language (NL)* documents by means of a comprehensive literature survey. To this end, we not only focus on existing techniques that have been adopted for SPLs, but also on the maturity of these approaches, that is, to what extent they could be applicable in practice. In particular, we make the following contributions:

- A comprehensive study of existing reverse engineering approaches, used input formats, employed NLP techniques and further algorithms for feature & variability extraction.
- A qualitative analysis regarding multiple criteria, such as accuracy, completeness, and their evaluation, which allow to compare the reviewed approaches at a reasonable level of detail.
- We provide key observations, identified within our detailed comparison, and derive shortcomings and challenges that are beneficial to identify future research directions.

2 NLP IN A NUTSHELL

NLP is a way for computers to analyze, understand, and derive meaning from human language. By utilizing NLP, developers can organize and structure knowledge to perform tasks such as automatic summarization, translation, or relationship extraction. We briefly introduce how NLP techniques are generally used, with a specific focus on feature identification and variability extraction in SPLs. The general process is shown in Figure 1.

As a first step, NL documents are transformed into types of words that can be identified and analyzed easily by computers. This step is called *Text Pre-processing*. In particular, NL documents are divided into words, phrases, symbols, or other meaningful elements (e.g., using tokenization). Additionally, these elements can be tagged with their type of word (e.g., noun, verb, object, etc.) using Part-of-Speech tagging (PoS tagging). In addition, stop words which usually refer to the most common words in a vocabulary (e.g., "the", "at") are removed in this phase, as they lack any linguistic information.

A second (optional) step is *Term weighting* which can be adopted to estimate the significance of terms in NL documents by calculating the frequency of their occurrence in different NL documents. For instance, Term Frequency-Inverse Document Frequency (TF-IDF) and C-NC value are two commonly used techniques. With TF-IDF, the term is considered important if it appears frequently in a document, but infrequently in other documents. C-NC value is a more sophisticated statistical measure which combines linguistic and statistical information.

Another optional step is *Semantic Analysis* that is typically used to gain semantic information. Several techniques can be employed in this step, for example, Vector Space Model (VSM) and Latent Semantic Analysis (LSA) are widely used to conduct a semantic analysis. With VSM, preprocessed documents are represented as vectors. Through calculating the cosine between the vectors, similarity between documents can be determined. LSA utilizes a term-document matrix to analyze the similarity of documents and can be combined with Singular Value Decomposition (SVD) to reduce the dimension of the textual documents.

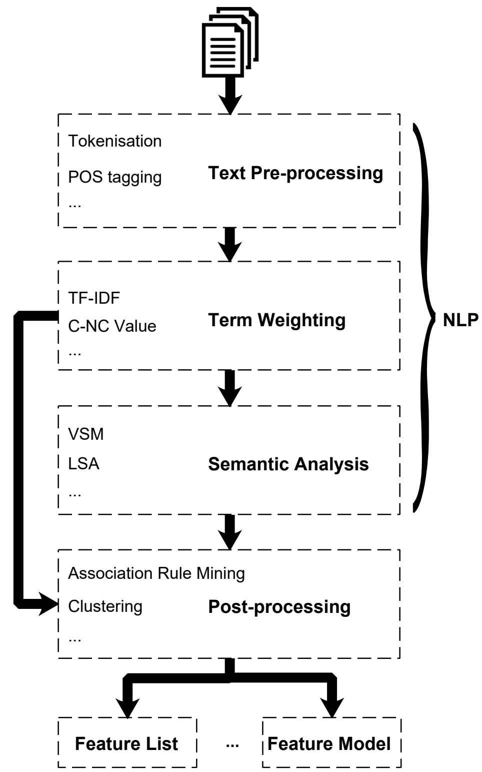

Figure 1: General process for applying NLP techniques in reverse engineering variability from NL documents.

Additionally to the NLP process, the transformed data can be further analyzed by a *post-processing* step in order to identify features and extract variability information. Certainly, various methods can be used in this step, such as, Clustering Approaches and Association Rule Mining. Cluster approaches are adopted to group similar features with a feature being a cluster of tight-related requirements. Association rule mining is used to discover affinities among features across products, and to augment and enrich the initial product profile. After post-processing, various outputs can be obtained, such as, a feature list or feature model.

3 REVIEW METHODOLOGY

A systematic literature review (SLR) is an accepted method for evaluating and interpreting all available research relevant to a particular research question, topic area, or phenomenon of interest [23]. In particular, we apply the proposed guidelines by Kitchenham et al. [23] in order to identify, classify, compare, and evaluate existing techniques for reverse engineering variability from natural language documents. In this section, we provide information about all steps we performed for planning the review.

3.1 Need for a Review

In recent years, the application of NLP techniques for aiding the software engineering process by reverse engineering information from several artefacts has been increased considerably. Our literature review aims to complement recent efforts by providing an overview of how NLP techniques are used to infer features from NL documents [6]. In particular, we extend the review by Bakar et al. by a) focussing also on how variability information is extracted and b) by a comprehensive qualitative evaluation of the approaches regarding aspects such as accuracy, automation, and tool support. As a contribution, we provide detailed insights for both, researchers and practitioners, in the current state and maturity of extracting detailed variability information from NL documents. Moreover, we not only highlight promising approaches, but also identify gaps and formulate derived challenges that have to be addressed in future, thus, paving the way for more efforts in this research directions.

3.2 Research Questions

The focus of this SLR is to identify, compare, and evaluate existing approaches for feature and variability extraction from NL documents. We formulate our research questions based on the PICOC method (Population, Intervention, Comparison, Outcome, and Context) [18] and present the respective criteria in Table 1.

While our overall question targets the applicability of current approaches in practice, we guide our systematic literature review by the following concrete research questions:

RQ1: What approaches of feature & variability extraction from NL documents have been proposed for SPLs?
With this question, we aim at summarizing all relevant techniques that contribute to the goal of our literature review. Although feature extraction is also subject to research in software engineering, we are mainly interested in approaches focussing on SPLs and how these approaches tackle the challenge of extracting variability information. Finally, we aim at identifying which kind of NL documents have been used as input for feature extraction.
RQ2: How are the techniques supported regarding tools and automation?
We are interested, whether the techniques, obtained through RQ1, are supported by robust tools or just implemented as prototypes. This is of special importance in the context of applicability in practice, such as in real-world systems or industry. Similarly, we evaluate to what extent the process of extracting features is automated.
RQ3: How reliable are the approaches, proposed for feature extraction in SPLs?
With this question, we focus on the quality of the proposed approaches. In particular, we are interested in two aspects. First, the completeness, that is, to what extent do the approaches also extract variability, thus, providing information for creating a complete picture of the SPL (e.g., by means of a feature model). Second, the accuracy, that is, how precisely do the proposed approaches extract features (and variability) from NL documents. As a result of this research question, we analyze to what extent the proposed approaches can be applied in practice or need to be revised and evaluated more thoroughly. Moreover, we derive limitations and open challenges from answering this research questions.

PICOC	Description
Population	Literature in reverse engineering variability from NL documents.
Intervention	Mechanisms, i.e., techniques, methods, tools, approaches that realize such a reverse engineering process.
Comparison	Techniques together with their performance, evaluation and tool support proposed by each primary study.
Outcome	Several observations regarding applicability and quality of current approaches and major gaps and open challenges in this field.
Context	The SPLE process, in particular the reverse engineering step to enable a systematic product-line development (e.g., in an extractive way).

Table 1: Research questions structured by PICOC criteria.

3.3 Search Strategy

To identify relevant literature and extract the important information, we set up a search strategy that consists of three steps. First, we specify scientific databases to search for our initial set of candidate papers. In particular, we search in the databases of ACM Digital Library, IEEE Xplore, SpringerLink, ScienceDirect, Scopus, dblp, and Google Scholar for studies published in journals, conferences, and workshops between the year 2000 and 2017. We have chosen these libraries as they are renowned scientific databases that index the most important publications in the field, such as from ACM, IEEE, or Elsevier. As a second step, we implement a review process to exclude duplicate studies or studies that are not relevant for other reasons (cf. Section 3.4). Afterwards, as a third step, we apply *snowballing* to complement the initial search [39]. In particular, we analyze references and citations of retrieved studies and secondary studies (i.e., existing surveys), thus, identifying relevant literature not covered by the aforementioned databases. Finally, we merge the results from our initial searching and snowballing to obtain the final set of *primary studies.*

3.4 Conducting the Review

Basically, we conduct our systematic review based on the protocol defined in the previous subsection. For instantiating this protocol, we have to take concrete actions as follows; *(i)* define the concrete search term, *(ii)* define inclusion and exclusion criteria for identifying relevant literature, and *(iii)* specify a concrete and systematic process for extracting the data needed to answer our RQs.

Search Criteria: To construct our search string, we derived keywords from our research questions based on the population, intervention, and outcome. Additionally, we checked for possible synonyms, related terms and alternative spellings. Finally, we used boolean logic; an ”OR” to combine alternative terms/spellings and an ”AND” to connect the major terms in our string. The resulting search string is as follows:

("feature extraction" OR "feature selection" OR "feature location" OR "feature identification" OR "feature detection" OR "feature mining" OR "feature clustering" OR "feature similarity") AND ("natural language" OR "requirement" OR "textual requirement" OR "description" OR "specification" OR "product review" OR "Natural language processing") AND ("Software Product Lines" OR "product family" OR "software family" OR "Feature-oriented software development")

In spite of the fact that all of the relevant keywords are of similar meanings, they differ somewhat. "Feature extraction" is the most frequent term in this topic used to describe the process of extracting features from NL documents. "Feature selection" usually means selecting a good feature set to achieve customer requirements, which focuses on the problem of optimization. "Feature location" predominantly concentrates on locating feature in source code. "Feature identification" and "feature mining" are both used as synonyms for feature extraction or feature location. "Feature detection" is highly relevant to detect dead features. "Feature clustering" refers to approaches where clustering algorithms have been employed to group features according to some clustering criteria. "Feature similarity" means the commonality of features. Understanding the small difference among these keywords, the study selection can be conducted more efficiently and comprehensively.

Inclusion and Exclusion Criteria: We created a set of *inclusion (IC) and exclusion (EC)* criteria to identify potential primary studies. While initially intended to be applied on the title and abstract, it turned out that this is insufficient for most of the papers to decide on their relevance. Hence, we also scanned introduction and conclusion to make a more reliable decision. The inclusion criteria we finally created, based on the analysis scope, are as follows:

- IC01 Articles matching the search string mentioned above and within the scope of our analysis, i.e., they propose a technique or mechanism for feature & variability extraction from NL documents.
- IC02 Articles published between January 1st 2000 to May 30th 2017, since research on automatic feature & variability extraction from NL documents in SPLs began in 21st century.

Moreover, we consider studies irrelevant if they meet at least one of the following exclusion criteria:

- EC01 Articles not focusing on feature and variability extraction from NL documents in SPL, i.e., feature extraction from legacy code, approaches improving feature modelling, functional requirements extraction, etc;
- EC02 Articles not written in English;
- EC03 Articles not belonging to research papers, i.e., proposals, summary of conference, lecture notes, etc;
- EC04 Articles not pertaining to the firsthand researches, namely, related literature review or survey papers.

Data Extraction: First of all, we applied our search string to the scientific databases. Afterwards, we applied inclusion and exclusion criteria to the result. Along with this process, we extracted and stored the following information in a spreadsheet:

- Date of search, scientific database, study identifier
- Publication information, author, title, publication year, source, publication type (Journal/Conference)
- Inclusion criteria *IC01–IC02* (and which one applies)
- Exclusion criteria *EC01–EC04* (and which one applies)

The first author initially applied the data extraction process. For all excluded paper, a brief reason was specified and double-checked by the second author. In case of non-agreement, the paper has been discussed by both authors to find a consensus. Once both authors finally decided on the primary studies, we performed a further retrieval by applying the snowballing method [39]. In particular, we took all primary studies and papers excluded due to EC04 into account. For papers, found by snowballing, the same extraction process as above was applied and if we decided on their relevance, it was added to the list of primary studies.

For the final list of primary studies, we read the full paper and captured and extracted additional data, required to perform our analysis and answer our research questions. In particular, we extracted the following data:

- *Input* and *Output* of the corresponding approach.
- *Methodology*, i.e., which NLP techniques are used and which further techniques are possibly applied in a post processing step. Moreover, we noted the degree of automation and how much of the variability (i.e., relation between features) can be recovered by a particular approach.
- *Evaluation:* to what extent a particular approach has been evaluated and what is the result of the evaluation, especially regarding accuracy
- *Tools:* Is any tool available, implementing the proposed approach and NLP techniques.

4 RESULTS

In this section, we present the results from our literature review. First, we briefly report on the results of our systematic literature search, described in Section 3. Then, we provide detailed answers to our formulated research questions.

4.1 Results of Studies Search

In order to obtain the primary studies, we initially used the predefined search string on the selected databases mentioned (cf. Section 3.3). As a result, we retrieved an initial list of relevant studies comprising 428 papers (ACM:22, IEEE:28, SpringerLink:102, ScienceDirect:5, Scopus:58, dblp:94, Google Scholar:119). Note that in this step, we already applied our inclusion criteria to the papers found, thus, all non-relevant papers have already been filtered out. Next, we applied our exclusion criteria to this initial list. After applying EC01, a majority of the papers could be discarded, as they propose relevant approaches, but on different artifacts than we are interested in. Further papers have been discarded, because they adhere to EC03 and EC04. Finally, we removed duplicated papers from our list, which eventually results into a list of nine papers.

Based on these selected papers and on the papers excluded due to EC04 (i.e., constituting secondary studies), we then performed *snowballing* as an additional step to retrieve relevant papers we may have missed so far. In particular, we performed three iterations of *backward* (i.e., analyzing the reference list of selected papers) and *forward* (i.e., screening papers on Google Scholar that cite our selected papers) snowballing [39]. In the first iteration, we found 445 papers (backward:384, forward:61), from which we excluded

Table 2: **Overview of all reviewed papers, ordered by year of appearance.**

ID	Title	Year	Source	Input	Accessibility	Output
P01	An Approach to Constructing Feature Models Based on Requirements Clustering	2005	[9]	SRS	Yes	FM
P02	An Exploratory Study of Information Retrieval Techniques in Domain Analysis	2008	[3]	SRS	No	FM
P03	A Framework for Constructing Semantically Composable Feature Models from Natural Language Requirements	2009	[38]	SRS	No	FM
P04	On-demand Feature Recommendations Derived from Mining Public Product Descriptions	2011	[13]	PD	Yes	FM
P05	Supporting Commonality and Variability Analysis of Requirements and Structural Models	2012	[25]	SRS	No	RTLs
P06	On Extracting Feature Models From Product Descriptions	2012	[1]	PD	Yes	FM
P07	Decision Support for the Software Product Line Domain Engineering Lifecycle	2012	[5]	PD	No	FM
P08	Mining Commonalities and Variabilities from Natural Language Documents	2013	[14]	PB	No	FL
P09	Supporting Domain Analysis through Mining and Recommending Features from Online Product Listings	2013	[17]	PD	Yes	FM
P10	Mining and Recommending Software Features across Multiple Web Repositories	2013	[40]	PD	Yes	FL
P11	Feature Model Extraction from Large Collections of Informal Product Descriptions	2013	[10]	PD	Yes	FM
P12	A Systems Approach to Product Line Requirements Reuse	2014	[30]	SRS	No	OVM
P13	Analyzing Variability of Software Product Lines Using Semantic and Ontological Considerations	2014	[33]	SRS	No	SRSS
P14	Generating Feature Models from Requirements : Structural vs . Functional Perspectives	2014	[19]	SRS	Yes	FM
P15	Detecting Feature Duplication in Natural Language Specifications when Evolving Software Product Lines	2015	[21]	SRS	No	DFs
P16	Improving the Management of Product Lines by Performing Domain Knowledge Extraction and Cross Product Line Analysis	2015	[34]	FD	Yes	FM
P17	Recommending Features and Feature Relationships from Requirements Documents for Software Product Lines	2015	[16]	SRS	Yes	FM
P18	Semantic Information Extraction for Software Requirements using Semantic Role Labeling	2015	[36]	SRS	No	SI
P19	CMT and FDE: Tools to Bridge the Gap between Natural Language Documents and Feature Diagrams	2015	[15]	PB	No	FM
P20	Automatic Semantic Analysis of Software Requirements Through Machine Learning and Ontology Approach	2016	[37]	SRS	No	SI
P21	Extracting Features from Online Software Reviews to Aid Requirements Reuse	2016	[7]	OR	Yes	FL
P22	Variability Analysis of Requirements: Considering Behavioral Differences and Reflecting Stakeholdersfi Perspectives	2016	[20]	SRS	No	FM
P23	Mining Feature Models from Functional Requirements	2016	[27]	SRS	No	FM
P24	Automated Extraction of Product Comparison Matrices from Informal Product Descriptions	2017	[29]	PD	Yes	PCM
P25	Extracting Software Features from Online Reviews to Demonstrate Requirements Reuse in Software Engineering	2017	[28]	OR	Yes	FL

SRS: software requirement specification; PD/PB: product description/brochure; OR: online software reviews; FD: feature diagrams.

FM: feature model; OVM: Orthogonal Variability Model; FL: feature list; RTLs: recommendation traceability links; SSRS: SRS similarity; DFs: duplicated features; SI: semantic information; PCM: product comparison matrix.

430 papers due to our exclusion criteria and removal of duplicates. Hence, 15 papers have been considered to be relevant, and thus, added to our list of primary studies. For these papers, we again applied the snowballing technique, resulting into 1191 papers, from which one paper remained after applying exclusion criteria and duplicate elimination. Finally, we also applied snowballing for the paper retrieved in the previous iteration, but from initially 58 potential papers all have been discarded (excluded or duplicated).

As a result, given our list of nine papers from the initial search, we were able to identify 16 further papers with snowballing that we found worth to be added to our primary studies. Hence, we found 25 papers as primary studies that are subject to further analysis in order to answer our research questions. We provide an overview of all papers, together with the extracted information, in Table 2 and answer our research questions in the following.

4.2 Answering Research Questions

The main goal of this SLR is to provide detailed insights of techniques used, degree of maturity achieved and shortcomings and challenges ahead for extracting feature and variability information from NL documents. To this end, we formulated three research questions in Section 3.2, which we answer in the following.

RQ1: What approaches of feature extraction from NL documents have been proposed for SPLs?
With this RQ, we shed light on techniques used for the extraction process and which kind of NL documents are considered as input for the respective approaches.
Techniques used: In Table 3, we provide a detailed overview of techniques used by the particular approaches. First of all, our analysis reveals that all of the phases, outlined in Section 2, have been employed by the considered approaches. For text pre-processing, the majority performs POS tagging (~ 44%) to decompose the documents in their building blocks. Moreover, tokenization was applied frequently (20%), yielding to a similar result. Beyond preprocessing, term weighting with different techniques is quite common amongst the approaches (~ 36%). Our inspection reveals that this technique is mainly used to identify meaningful terms that resemble possible features. An interesting observation we made, is that most approaches employ more than one technique, either from preprocessing only or in combination with term weighting.

Observation 1. *For extracting features and variability, a diverse selection of NLP techniques is employed, indicating that a) different approaches may lead to the desired result and b) applying multiple NLP techniques increases the quality of the result.*

While above-mentioned approaches focus mainly on syntactical aspects, many approaches also make use of the (optional) built-in mechanisms of NLP for semantic analysis (~ 56%). In particular, latent semantic analysis (LSA) and semantic role labeling are preferred techniques, where the latter is combined with using a proper lexical database. If we take quality criteria such as accuracy or completeness into account (cf. Table 4), we observe a slight tendency that an additional semantic analysis improves the overall result.

Observation 2. *Semantically understanding the NL documents is a crucial aspect for successfully reverse engineering features & variability.*

Finally, many approaches (~ 60%) perform an additional post-processing step on the top of previously mentioned NLP mechanisms (cf. Figure 1). Our analysis reveals that especially clustering algorithms are preferably used in this stage of the extraction process (~ 32%). The reason is that clustering is able to find relations between previously identified concept or terms. Hence, such algorithms are especially beneficial to finding related and unrelated features, and thus, enable the search for groups of features and even other relationships among them such as hierarchies.

Observation 3. *Clustering algorithms are well-suited to complement the NLP process, as they facilitate the detection of specific relationship between features such as groups or parent-child relations.*

Besides clustering, a variety of other techniques are employed for post-processing (~ 48%), mainly from the domain of machine learning. The particular algorithms we identified are especially able to learn specific pattern in the preprocessed and enriched data from the NLP process. In this way, even complex dependencies can be inferred, which is hardly possible with NLP mechanisms only.
Types of documents: As NL documents may occur in various forms, we are also curious about which kind of documents are subject to feature/variability extraction. According to Table 2, our review indicates that software requirements specifications (SRS) are used predominantly (> 50%) as input, usually based on standards such as IEEE-STD-830. Interestingly, the structure (e.g., hierarchies) of such documents is only rarely employed, though it may contain valuable information such as for grouping features or establishing parent-child relationships. Moreover, only few data sets for SRS are provided, thus, preventing from replicating most of the studies. Next frequently, product descriptions (PD) are used (~ 28%), mainly due to their availability (e.g., via web pages such as softpedia.com) and the rich information, they contain. For instance, features are likely to appear more explicit in such documents and sometimes even bullet lists are provided. Finally, product brochures (PB), i.e., documents used for marketing reasons, are sometimes used. Usually, they highlight main features of a product, thus, making it easy to extract them. However, due to their limited purpose (e.q, acquiring new customers), they are usually incomplete and contain only few or no variability information.

Observation 4. *Software requirements specifications are frequently used as input for feature & variability extraction from NL documents. However, the reviewed papers provide no or only limited (i.e., small, artificial) SRS documents, thus, impeding replicability and making applicability in practice questionable.*

RQ2: How are the techniques supported regarding tools and automation?
Since the process of extracting features & variability is pretty complex and tedious, sophisticated tool support and a high degree of automation are inevitable. We use both aspects as quality criteria for the reviewed papers, and thus, evaluate them using a 3-point scale. The results are presented in Table 4 (last two columns), with tool support further divided into tools for NLP which is indispensable on the basis of Figure 1 and tools for feature & variability extraction (FVE) developed based on the proposed approaches.
For tool support, ● means that comprehensive tool support exists *and* is available. Accordingly, ◐ indicates that tool support is described, but not available. Finally, if tool support is neither provided nor described, the approach is rated with ○. We use the same symbols for the degree of automation, indicating that an approach is fully automated, semi-automated, or only minor/not automated.
Tool support: Our analysis reveals that many approaches do not provide any tool support (~ 52% for NLP & ~ 36% for FVE). In most of these cases, only algorithms used are mentioned, but neither their origin nor how multiple algorithms are put together is elaborated. This, in turn makes it hard to reason about the respective approaches, thus, mitigating the trust in the applicability of them. For another, fairly large amount of approaches (~ 32% for NLP & ~ 48% for FVE), tools are described, but not provided or not available anymore (e.g., [38]). This is especially surprising, as in some of these cases researchers invested considerable effort in building whole frameworks to automate the extraction process.

Table 3: Overview of the techniques used by the reviewed approaches in the particular phases of the extraction process.

Techniques			ID
NLP Techniques	Text Pre-processing	Tokenisation	P06 P09 P12 P15 P19
		Part of Speech Tagging	P06 P08 P09 P10 P12 P15 P18 P19 P20 P24 P25
		Lemmatization	P24 P25
		Stemming	P11 P25
	Term Weighting	TF-IDF	P04 P09 P10 P11 P23 P25
		C-NC Value	P08 P19 P24
	Semantic Analysis	Vector Space Model	P02 P05
		Latent Semantic Analysis	P02 P03 P21 P22 P25
		Contrastive Analysis	P08 P21 P24
		Syntactical Heuristics	P24
		Latent Dirichlet Allocation	P10
		Semantic Role Labeling	P13 P14 P18 P20 P22
		Semantic Model	P23
	Lexical Database	WordNet	P07 P16 P20 P23
		SemCor	P20
		PropBank	P18 P20
		FrameNet	P18 P20
Clustering Techniques	Partitional Clustering	K-Means	P10 P21
		K-Medoids	P10
		SPK-Means	P04 P11
	Fuzzy Clustering	Fuzzy C-Means	P09 P21
	Neural Clustering	Self Organizing Map Clustering	P21
	Hierarchical Clustering	Hierarchical Agglomerative Clustering	P01 P02 P16
	Incremental Clustering	Incremental Diffusive Clustering	P04 P09 P10
Further Techniques	Greedy Algorithm		P20
	Formal Concept Analysis		P23
	Propositional Logic		P06
	Association Rule Mining		P09 P10 P11
	Heuristics		P17 P23
	Decision Tree		P18
	K-nearest Neighbors		P04 P19 P20
	Edmondsfi Algorithm		P11 P16
	Maximum Entropy Method		P20

Even worse, the non-availability also questions the sustainability of such approaches, that is, whether they have been created for practical usage or just for theoretical evaluations. Finally, only for a minor amount (~ 16% for both, NLP & FVE), tools are available online, for instance on Github, usually complemented by examples and further material. In some cases, the tools appear to be mature, stable, and designed to be used by a wide range of researchers and practitioners (e.g., [15, 29]).

Observation 5. *When tools are available, they make a stable, maintained, and reliable impression,and thus, are likely to be applicable in practice. However, in most cases, no tools are described or even exist, thus, mitigating trust in applicability and reliability of the corresponding approaches.*

Automation: The vast majority of approaches (> 90%) foster a semi-automated approach, where at least some manual adjustment is required. Most commonly, parameters and thresholds need to be specified (e.g., [1, 13, 38]) or domain analysts have to interact with the described tool in order to correct or validate information. Approaches that provide full automation refrain from taking user input into account, although this may influence the results, e.g., regarding feature names. Moreover, our analysis reveals that these approaches usually exhibit a rather low accuracy (cf. Table 4).

Observation 6. *Most approaches provide a high degree of automation, whereas full automation is an exception (but possible in general). This is mainly due to the complex extraction process, which requires manual assessment and domain knowledge to achieve a satisfiable result.*

RQ3: How reliable are the approaches, proposed for feature extraction in SPLs?
With this research question, we investigate the result quality of the extraction process. In particular, we focus on two quality criteria; *accuracy*, that is, to what extent the identified feature & variability information is correct; and *completeness*, that is, whether and to what extent feature & variability information is extracted. Moreover, we assess how comprehensive the approaches have been evaluated.

Table 4: Result of the qualitative analysis of primary studies.

ID	Acc.	Compl.	Eval	Tool Support		Automation
				NLP	FVE	
P01	○	◐	◐	○	○	◐
P02	○/◐	○	○	◐	◐	◐
P03	○/◐	◐	○	◐	◐	◐
P04	◐	◐	◐	○	○	◐
P05	○	○	○	○	○	◐
P06	◐	◐	◐/●	○	●	◐
P07	◐	◐	◐	●	●	◐
P08	○	◐	○	○	○	◐
P09	◐	◐	◐/●	◐	◐	◐
P10	◐	○	○/◐	○	◐	◐
P11	◐	◐/●	○/◐	○	○	●
P12	◐	◐	◐	○	○	◐
P13	●	○	◐	○	◐	◐
P14	○	●	○/◐	○	◐	◐/●
P15	○	○	○	●	○	◐
P16*	○	○	○	○	○	●
P17	○	◐	○	●	◐	◐
P18	○	○	○	○	○	◐
P19	○	◐	○	●	●	◐
P20	○	○	◐	◐	◐	◐
P21	◐	○/◐	◐	◐	◐	◐
P22	○	●	◐	○	◐	◐/●
P23	◐	○/◐	◐	◐	◐	◐
P24	◐	○	◐/●	◐	●	◐
P25	○	◐	○	◐	◐	◐

FVE: feature & variability extraction; *this approaches has been negatively evaluated, because it takes already existing feautre diagrams as input, thus, being partially out of scope.

We present the results in Table 4, using the same 3-point scale as in the previous RQ, having the following meaning. For *accuracy*, the ● means that the approach is very accurate, i.e., features (and variability) are reverse engineered correctly. Accordingly, the ◐ means that the approach is sufficiently accurate, that is, main information is correct but contains minor inconsistencies (e.g., wrong/missing features or variability). Finally, the ○ means that the approach is inaccurate, thus, only capable for providing a very high level overview of separated concerns. For *completeness*, the ● means that the complete information (i.e., features & variability) is extracted explicitly, most likely in form of a feature model. In contrast, the ◐ indicates that variability information is only partially extracted. Consequently, the ○ means that no variability information can be extracted by the approach, thus, the result is only a list of features. For the *evaluation*, ● indicates that a comprehensive and also *reproducible* revaluation is provided that allows for a detailed reasoning about the proposed approach. Likewise, the ◐ means that the evaluation has some limitations, for instance, only a small or artificial case study is used or the case study is not reproducible, thus, the results of the study can not be verified. Finally, the ○ refers to approaches that provide no or only a weak evaluation.

Accuracy: Our analysis reveals that many approaches (∼ 56%) are inaccurate, however, for various reasons. While some of them perform rather bad in corresponding metrics (e.g., precision, recall; [9, 16, 21]), other approaches simply do not provide any information about it. For the latter, the reason is that they focus on other aspects such as usefulness for guiding developers, thus, the evaluation does not provide any quantitative measures (e.g., [19, 20]). Next, some approaches (∼ 40%) provide relatively accurate results, which can be seen as a starting point for further, manual refinement. Finally, only one approach (∼ 4%) is highly accurate, and thus, can be used out-of-the-box (i.e., without manual adjustment).

Observation 7. *The accuracy is one of the most critical problems that needs to be improved for achieving practical applicability. However, it seems that even with less or unknown accuracy, several approaches perform well in supporting developers in manual domain analysis or even extraction processes.*

Completeness: Generally, many approaches extract partial variability information (∼ 48%). However, in most of these cases, only some relations (e.g., mandatory, optional) are extracted, and thus, important information is missing. For a similar amount of approaches, only features are identified, but not their relationships (∼ 44%), which means that they are of limited usefulness for the SPLE process. Finally, only a minor proportion (∼ 8%) provides complete and explicit variability information that makes it possible to generate a complete feature model with detailed relationships.

Observation 8. *When features and variability information are complete and explicit, they constitute a feature model with detailed relationships. However, in most cases, the approaches are incomplete wrt. variability information, thus, requiring increased manual effort for recreating this information.*

Evaluation: Surprisingly, many of the reviewed approaches (∼ 52%) provide a weak or even no evaluation. In some cases, the approaches are just sketched, but fail to evaluate it in any sense, while others lack important information such as study design or sound evaluation criteria. Furthermore, certain approaches (∼ 36%) provide a basic evaluation, but lack reproducibility due to missing access to tools and/or data sets. Finally, only a minor proportion (∼ 12%) presents a comprehensive and reproducible evaluation that gives valuable insights in benefits and limitation of the respective approaches, thus, making their claimed contribution convincing.

Observation 9. *A comprehensive and reproducible evaluation makes an approach more reliable and allows for reproducibility. However, in most cases, evaluations are performed either in a weak and unsound manner or lack important resources to reproduce them. Especially the latter aspect impedes an objective comparison.*

4.3 Threats to Validity

Construct Validity: Our search string may be incomplete, and thus, limit the diversity of information from digital engines. We addressed this issue by diversifying search terms, extending to their synonyms, and elaborating the different meanings between them. In addition, we carefully derived the search term based on our research questions.

Internal Validity: First, we may have overseen important papers to be included in the survey, due to bias of primary study selection or negligence. We addressed this issue by an independent repetition of the literature search by a second author, according to the

presented methodology (Section 3). Second, the assessment of the approaches, regarding our proposed quality criteria, is prone to be subjective, thus, introducing bias in the overall evaluation. We addressed this issue as follows: Two authors of this paper *independently* assessed all primary studies regarding the criteria given in Table 4. Afterwards, theses authors compared their results; in case their opinion diverged, they discussed reasons for their assessment and, finally, made a common decision.
External Validity: We didn't expand the primary study selection to books, which possibly affects the generalizability of our study.
Conclusion Validity: The data extraction may be of bias. To tackle it, the first author initially extracted data complying with the predefined data extraction form, then double-checked by the second author and discussed whether the data was accurate and appropriate to answer the research questions.

5 DISCUSSION

Based on our detailed analysis, we summarize and discuss our observations. In particular, we elaborate on aspects that constitute challenges and need to be improved in future research.
Input format matters. In our survey, we found SRS and product descriptions to be the most common input for feature & variability extraction. While SRS provide the most detailed information, representing the domain, we identified a lack of access to SRS, which impedes the progress in developing approaches for information extraction from such documents. On the other hand, product descriptions are freely available, but only reflect an incomplete overview of a domain. For future research, we see two challenges. First, a detailed comparison (by means of a sound evaluation) to what extent product descriptions can be used for domain analysis as an alternative for SRS. Second, to design reverse engineering approaches so that they can be used flexibly with different input formats, in particular, supporting SRS and product descriptions.
Extracting variability is challenging. Extracting variability information is by far the most challenging task, indicated by the rather low proportion of accurate and complete approaches. Depending on the kind of input documents, different approaches are used to extract variation points. However, most of them need manual intervention, i.e., the result of the automated extraction is not accurate and complete enough to get rid of domain analysts' correcting. The reason is that variability extraction from NL documents is a process of understanding natural language, which is usually full of ambiguities. Thus, the challenge is to improve existing approaches, either by new combinations of existing techniques or by developing new techniques. Also, we see great potential in taking additional information (e.g., domain knowledge) into account, which poses the challenge of integrating it in an automated extraction process.
Rethinking sustainability. Tool support and a coherent evaluation are mostly missing, due to several reason. For making progress in extracting features & variability, especially regarding its applicability in practice, these aspects need to be addressed in future. First, establishing a ground truth (e.g., a gold standard) is inevitable for assessing future approaches. This not only allows to compare approaches with each other, but also to draw conclusion about their performance (in terms of accuracy) and robustness. Second, reusing existing approaches for reproducibility and improvement is of superior importance for making the next step. This, in turn, requires a common sense of tool building, which may go beyond the scope of pure research. Nevertheless, we argue that one of the challenges is that fundamental ideas are backed by tools that can be accessed by others. This way, researcher can join forces, and thus, push the boundaries for extracting features & variability.

6 RELATED WORK

In this section, we discuss related literature reviews addressing reverse engineering techniques for extracting feature and analyzing the commonality and variability of products.

Dit et al. conducted a systematic review of feature location techniques in source code, including case studies and tools, and then, presented a taxonomy for classification along nine key dimensions [11]. However, although encompassing NLP and Information Retrieval (IR) techniques, the considered techniques address source code comments, identifiers, etc., rather than textual requirements.

Khurum and Gorschek presented a systematic review of domain analysis solutions to analyze the level of industrial application and/or empirical validation of the proposed solutions [22]. Moreover, they investigate usability and usefulness of proposed approaches. However, this review does not present the specific approaches, used to identify features and their relationships from the primary studies, especially targeting NL documents.

Lisboa et al. conducted a systematic review of domain analysis tools which support the domain analysis process to identify and document common and variable characteristics of systems in a specific domain [26]. This review covers a large-scale tools analyzing different sources, instead of only focusing on NL documents, thus being less comprehensive.

Berger et al. presented a systematic review of variability modeling practices in industrial SPLs to provide insights into application scenarios and perceived benefits of variability modeling, notations and tools used, the scale of industrial models, and experienced challenges and mitigation strategies [8]. However, this review focuses on notations and related tools employed in variability modelling rather than on approaches for extracting of features & variability.

Alves et al. conducted a systematic review of requirements engineering for SPLs to suggest important implications for practice, and identifying research trends, open problems, and areas for improvement [2]. This review also provides a survey of semi-automatic or automatic tools used. However, it was conducted in 2009, leading to the lack of the latest tools. The types of surveyed requirements artifacts include not only NL documents, but also requirements in various forms, e.g., features and orthogonal variability models. In addition, it does not focus on approaches for feature & variability extraction.

Bakar et al. presented a systematic review of feature extraction approaches from NL requirements for reuse in SPLs [6]. This review provides a detailed survey of approaches used for identifying features and analyzing their relationships from textual requirements, e.g., NLP techniques and clustering approaches, and also specifies the evaluation approaches. However, this review just contains 13 studies, which can not offer a comprehensive survey and doesn't provide sufficient evidences of the available tools supporting the variability information extraction.

7 CONCLUSION

Feature and variability information extraction from NL documents can obtain a explicit mapping of feature and variability information to other artifacts. However, there are few systematic literature reviews providing a comprehensive and detailed survey of approaches and tools used for extracting features and their relationships from NL documents. In this paper, we present the results of our systematic literature review of 29 papers that propose approaches to extract features and variability and which we compared and analyzed qualitatively based on several criteria.

Based on our review, we made several observations and derived implications and challenges for current but also future research in this field. Among other, our key findings are that
a) software requirements and product descriptions are the most common NL documents, but exhibit differences with respect to the information used for extraction;
b) Many approaches are neither accurate nor complete, and thus, of limited use in practice due to increased manual effort or simply wrong information
c) tool support is rather sparse, which impedes reproducibility, and thus, makes a fair comparison and reasoning impossible
d) Full automation is hard to achieve and, based on several evaluations, may even not be wanted or possible, because some amount of domain knowledge is only manually available or expressible.

Based on our findings, we suggested several actions to overcome current limitations, but also to provide more solid foundations for future research in this direction. As future work, we intend to tackle some of the mentioned challenges by ourselves.

REFERENCES

[1] M. Acher, A. Cleve, G. Perrouin, P. Heymans, C. Vanbeneden, P. Collet, and P. Lahire. 2012. On Extracting Feature Models From Product Descriptions. In *Proc. Int'l Workshop on Variability Modeling of Software-intensive Systems.* ACM, 45–54.

[2] V. Alves, N. Niu, C. Alves, and G. Valença. 2010. Requirements Engineering for Software Product Lines: A Systematic Literature Review. *Inf. and Soft. Technology* 52 (2010), 806–820.

[3] V. Alves, C. Schwanninger, L. Barbosa, A. Rashid, P. Sawyer, P. Rayson, C. Pohl, and A. Rummler. 2008. An Exploratory Study of Information Retrieval Techniques in Domain Analysis. In *Proc. Int'l Software Product Line Conference.* IEEE, 67–76.

[4] S. Apel, D. Batory, C. Kästner, and G. Saake. 2013. *Feature-Oriented Software Product Lines.* Springer.

[5] E. Bagheri, F. Ensan, and D. Gasevic. 2012. Decision Support for the Software Product Line Domain Engineering Lifecycle. *Autom. Soft. Eng.* 19 (2012), 335–377.

[6] N. H. Bakar, Z. M. Kasirun, and N. Salleh. 2015. Feature Extraction Approaches From Natural Language Requirements for Reuse in Software Product Lines: A Systematic Literature Review. *J. Sys. and Soft.* 106 (2015), 132–149.

[7] N. H. Bakar, Z. M. Kasirun, N. Salleh, and H. A. Jalab. 2016. Extracting Features from Online Software Reviews to Aid Requirements Reuse. *J. Applied Soft Computing* 49 (2016), 1297–1315.

[8] T. Berger, R. Rublack, D. Nair, J. M. Atlee, M. Becker, K. Czarnecki, and A. Wąsowski. 2013. A Survey of Variability Modeling in Industrial Practice. In *Proc. Int'l Workshop on Variability Modeling of Software-intensive Systems.* ACM, 1.

[9] K. Chen, W. Zhang, H. Zhao, and H. Mei. 2005. An Approach to Constructing Feature Models Based on Requirements Clustering. In *Proc. Int'l Conf. Requirements Engineering.* IEEE, 31–40.

[10] J. Davril, E. Delfosse, N. Hariri, M. Acher, J. Cleland-Huang, and P. Heymans. 2013. Feature Model Extraction from Large Collections of Informal Product Descriptions. In *Proc. Europ. Software Engineering Conf./Foundations of Software Engineering.* ACM, 290–300.

[11] B. Dit, M. Revelle, M. Gethers, and D. Poshyvanyk. 2013. Feature Location in Source Code: A Taxonomy and Survey. *J. of Softw.: Evolution and Process* 25 (2013), 53–95.

[12] Y. Dubinsky, J. Rubin, T. Berger, S. Duszynski, M. Becker, and K. Czarnecki. 2013. An Exploratory Study of Cloning in Industrial Software Product Lines. In *Proc. Eur. Conf. Soft. Maint. and Reeng.* IEEE, 25–34.

[13] H. Dumitru, M. Gibiec, N. Hariri, J. Cleland-Huang, B. Mobasher, C. Castro-Herrera, and M. Mirakhorli. 2011. On-demand Feature Recommendations Derived from Mining Public Product Descriptions. In *Proc. Int'l Conf. Software Engineering.* ACM, 181.

[14] A. Ferrari, G. O. Spagnolo, and F. Dell'Orletta. 2013. Mining Commonalities and Variabilities from Natural Language Documents. In *Proc. Int'l Software Product Line Conference.* ACM, 116.

[15] A. Ferrari, G. O. Spagnolo, S. Gnesi, and F. Dell'Orletta. 2015. CMT and FDE: Tools to Bridge the Gap Between Natural Language Documents and Feature Diagrams. In *Proc. Int'l Software Product Line Conference.* ACM, 402–410.

[16] M. Hamza and R. J. Walker. 2015. Recommending Features and Feature Relationships from Requirements Documents for Software Product Lines. In *Proc. Int'l Workshop Realizing Artificial Intelligence Synergies in Soft. Eng.* IEEE, 25–31.

[17] N. Hariri, C. Castro-Herrera, M. Mirakhorli, J. Cleland-Huang, and B. Mobasher. 2013. Supporting Domain Analysis through Mining and Recommending Features from Online Product Listings. *IEEE Trans. Soft. Eng.* 39 (2013), 1736–1752.

[18] Julian PT Higgins and Sally Green. 2011. *Cochrane Handbook for Systematic Reviews of Interventions.* Vol. 4. John Wiley & Sons.

[19] N. Itzik and I. Reinhartz-Berger. 2014. Generating Feature Models from Requirements: Structural vs. Functional Perspectives. In *Proc. Int'l Software Product Line Conference.* ACM, 44–51.

[20] N. Itzik, I. Reinhartz-Berger, and Y. Wand. 2016. Variability Analysis of Requirements: Considering Behavioral Differences and Reflecting Stakeholders' Perspectives. *IEEE Trans. Soft. Eng.* 42 (2016), 687–706.

[21] A. Khtira, A. Benlarabi, and B. El Asri. 2015. Detecting feature duplication in natural language specifications when evolving software product lines. In *Proc. Int'l Conf. Evaluation of Novel Approaches to Software Engineering.* IEEE, 257–262.

[22] M. Khurum and T. Gorschek. 2009. A Systematic Review of Domain Analysis Solutions for Product Lines. *J. Sys. and Soft.* 82 (2009), 1982–2003.

[23] B.A. Kitchenham. 2007. Guidelines for performing systematic literature reviews in software engineering. In *EBSE Technical Report.*

[24] Charles W. Krueger. 2001. Easing the Transition to Software Mass Customization. In *Proceedings of the International Workshop on Software Product-Family Engineering.* Springer, 282–293.

[25] K. Kumaki, R. Tsuchiya, H. Washizaki, and Y. Fukazawa. 2012. Supporting Commonality and Variability Analysis of Requirements and Structural Models. In *Proc. Int'l Software Product Line Conference.* ACM, 115–118.

[26] L. B. Lisboa, V. C. Garcia, D. Lucrédio, E. S. de Almeida, S. R. de Lemos Meira, and R. P. de Mattos Fortes. 2010. A Systematic Review of Domain Analysis Tools. *Information and Software Technology* 52 (2010), 1–13.

[27] H. Ben-Abdallah M. Mefteh, N. Bouassida. 2016. Mining Feature Models from Functional Requirements. *Comput. J.* 59 (2016), 1784–1804.

[28] N. Salleh N. H. Bakar, Z. M. Kasirun and A. H. Halim. 2017. Extracting Software Features from Online Reviews to Demonstrate Requirements Reuse in Software Engineering. In *Proc. Int'l Conf. Computing and Informatics.* Sintok: School of Computing, 184–190.

[29] S. B. Nasr, G. Bécan, M. Acher, J. B. Ferreira Filho, N. Sannier, B. Baudry, and J. Davril. 2017. Automated Extraction of Product Comparison Matrices from Informal Product Descriptions. *J. Sys. and Soft.* 124 (2017), 82–103.

[30] N. Niu, J. Savolainen, Z. Niu, M. Jin, and J. R. C. Cheng. 2014. A Systems Approach to Product Line Requirements Reuse. *IEEE Systems Journal* 8 (2014), 827–836.

[31] L. M. Northrop P. C. Clements. 2001. *Software Product Lines: Practices and Patterns.* Addison-Wesley Professional.

[32] K. Pohl, G. Böckle, and F. van Der Linden. 2005. *Software Product Line Engineering: Foundations, Principles and Techniques.* Springer.

[33] I. Reinhartz-Berger, N. Itzik, and Y. Wand. 2014. Analyzing Variability of Software Product Lines Using Semantic and Ontological Considerations. In *Proc. Int'l Conf. Advanced Information Systems Engineering.* Springer, 150–164.

[34] I. Reinhartz-Berger and O. Wulf-Hadash. 2015. Improving the Management of Product Lines by Performing Domain Knowledge Extraction and Cross Product Line Analysis. *Inf. and Soft. Technology* 59 (2015), 191–204.

[35] C. Riva and C. Del Rosso. 2003. Experiences with Software Product Family Evolution. In *Proc. Int'l Workshop on Principles of Software Evolution.* IEEE, 161–169.

[36] Y. Wang. 2015. Semantic Information Extraction for Software Requirements using Semantic Role Labeling. In *Proc. Int'l Conf. Progress in Informatics and Computing.* IEEE, 332–337.

[37] Y. Wang. 2016. Automatic Semantic Analysis of Software Requirements Through Machine Learning and Ontology Approach. *J. Shanghai Jiaotong University* 21 (2016), 692–701.

[38] N. Weston, R. Chitchyan, and A. Rashid. 2009. A Framework for Constructing Semantically Composable Feature Models from Natural Language Requirements. *Proc. Int'l Software Product Line Conference* (2009), 211–220.

[39] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén. 2012. *Experimentation in Software Engineering.* Springer Science & Business Media.

[40] Y. Yu, H. Wang, G. Yin, and B. Liu. 2013. Mining and Recommending Software Features across Multiple Web Repositories. In *Proc. Asia-Pacific Symposium on Internetware.* ACM, 1–9.