Databases and Information Systems X
A. Lupeikiene et al. (Eds.)

doi:10.3233/978-1-61499-941-6-109

Perspectives of Information Requirements Analysis in Big Data Projects

Natalija KOZMINA[1], Laila NIEDRITE and Janis ZEMNICKIS
Faculty of Computing, University of Latvia,
Riga, Latvia

Abstract. Big data technologies are rapidly gaining popularity and become widely used, thus, making the choice of developing methodologies including the approaches for requirements analysis more acute. There is a position that in the context of the Data Warehousing (DW), similar to other Decision Support Systems (DSS) technologies, defining information requirements (IR) can increase the chances of the project to be successful with its goals achieved. This way, it is important to examine this subject in the context of Big data due to the lack of research in the field of Big data requirements analysis. This paper gives an overview and evaluation of the existing methods for requirements analysis in Big data projects. In addition, we explore solutions on how to (semi-) automate requirements engineering phases, and reason about applying Natural Language Processing (NLP) for generating potentially useful and previously unstated information requirements.

Keywords. Big data, requirement analysis, literature review, NLP

1. Introduction

Big data usually is understood as large and complex data that needs new computer technologies and new methods for processing. The large amount of data is not the only property that defines the necessity for new techniques. The most popular definition of Big data [1] includes 3 Vs: Volume, Variety, and Velocity. These Vs characterize the large amount and the complexity of data as well as the data generation speed that causes the necessity to deal with the streaming data. Other Vs added later to the list, e.g. Veracity and Value, describe the uncertainty and business value of Big data.

Big data is involved in many human everyday activities, it is also connected with many other widely known terms e.g. social networks, Internet of Things (IoT), Big data analytics, cloud computing, NoSQL, etc. that help to understand how the Big data is emerging or how it can be processed.

Currently many attempts are made to develop Big data solutions having different level of success. Projects' failures are often caused by the problems [2] that are determined by the Big data features, e.g. finding the best way how to extract the value from Big data, integration problems of Big data sources, or data quality problems. Research provides also more technological Big data issues [3]: storage and processing issues, analytical, and technical challenges. Existing storage capabilities do not satisfy

[1] Corresponding author, Natalija Kozmina, Faculty of Computing, University of Latvia, Raina blvd. 19, Riga, Latvia; E-mail: natalija.kozmina@lu.lv.

the needs to store large amounts of data produced by different sources, e.g. social media or sensor devices. Processing of big data volumes is also time-consuming. Among the analytical issues the following questions are important: do we really need to store and to analyze all big data volumes, and how to choose the data elements that provide the most valuable information. The analysis methods should be applied carefully depending on the expected results. Regarding the technical challenges researchers have named the following issues [3]: fault tolerance, scalability, data quality problems, and heterogeneous data. Fault-tolerant computing should ensure acceptable time period, but in the case of big data that can be done only with highly complex algorithms. The scalability issue of big data can be solved by cloud computing with high level of resource sharing cost-effectively. Despite the large volumes of Big data, the focus is on storing relevant and qualitative data, therefore, it is necessary to determine, which data is relevant, or how to determine the level of accuracy of data needed for decision making. In most cases, Big data is unstructured, which is hard and costly to process; it is not always possible to convert it into structured data that can be automatically processed.

Problems of Big data projects' failures [4] can be solved by the right choice of tools and methods, for example, automation and agile techniques [5]. It is also important that the appropriate data is collected according to the *information requirements* (IR) of the company. IR define information, which should be available after the development of the information system is finished.

This paper gives an overview of the existing methods associated with Big data technology and requirements analysis, and provides their evaluation by three types of criteria: (i) general characteristics, (ii) requirements analysis related, and (iii) Big data technologies related criteria.

The structure of the paper is as follows: Section 2 provides a detailed description of the literature review process, in Section 3 we elaborate on the review results and categorize them, Section 4 is dedicated to the discussion on information requirements definition and elicitation from Big data, and Section 5 concludes the paper.

2. The Literature Review Process

In this section we describe the literature review process that was conducted according to the most commonly used guidelines given by Kitchenham and Charters [6]. The literature review process is composed of three major phases: (i) preparing for the review, (ii) conducting the review process, and (iii) reporting the results of the study.

2.1. Preparing for the Review

The goal of our literature review was to explore the aspects of information requirements analysis in the context of Big data.

We base our study on the following research questions:

RQ1. How the requirement analysis applied in the context of Big data?

RQ2. What empirical methods have been applied for the requirement analysis in the field of Big data?

RQ3. Is it feasible to generate the information requirements (IR) in a Big data project by processing the existing data in a (semi-) automatic way?

2.2. Conducting the Review Process

2.2.1. Source Selection Strategy and Search Queries

We performed search in 6 widely used electronic publication databases: ACM, Scopus, IEEE Xplore, SpringerLink, Web of Science, and Google Scholar.

A Google Scholar search query was as follows: *<"Requirement engineering" AND "Big Data" AND ("Analysis Requirements" OR "Information Requirements" OR "Data Requirements")>*. Search queries for other databases could slightly differ due to peculiarities of each particular database, but the semantic meaning stayed the same.

2.2.2. Inclusion (IC) and Exclusion Criteria (EC)

A study was selected for further detailed analysis, if it met all the IC and did not cover any of the EC. All the IC and EC are summarized in Table 1. These criteria would ensure that the results of the survey are the most relevant and in line with research questions stated in Section 2.1.

Table 1. Inclusion (IC) and exclusion (EC) criteria applied at a certain analysis stage.

ID	Criteria description	Stage
IC1	The publication date falls into the time interval from 2014 to 2017	1
IC2	The study is indexed in at least one of the selected publication databases	1
IC3	The language of the study is English	1
IC4	The title, keywords, and abstract of the study is related to at least one of the formulated research questions	1
IC5	The contents (headings, figures, table, introduction, and conclusions) of the study is related to at least one of the formulated research questions	2
EC1	A duplicate paper	2
EC2	In case of multiple versions of the paper, only one is included (either the latest or the fullest)	2
EC3	The study is a (PhD, Master, or Bachelor) thesis, a survey or a literature review, a poster paper, a standard, a books, a tutorial	2
EC4	The keyword "Big data" is not present in the paper body (e.g. in the Reference section only)	3
EC5	The approach presented in the study is too specific and cannot be generalized to be applied in other domains (e.g. a study on geospatial data or electromagnetic inference)	3

2.2.3. Paper Selection and Search Results in Numbers

The process of paper selection included 3 stages, during which we subsequently applied IC and EC (see Table 1) to the list of literature studies. A schematic representation of the paper selection process is depicted in Figure 1. Throughout all the paper selection process we applied the *Quality Assessment Criteria* to ensure the quality and relevance of the studies complies with the goal of our research:

Q1. Is the contribution of the paper related to requirement analysis in Big data?

Q2. Is the contribution of the paper novel and it covers any improved methodology or open issue?

Q3. Is there any case study or approbation example for the methodology proposed in the paper?

Q4. Is the theoretical approach or approbation example stated clearly?

Each study received an assessment of Q1-Q4 as a numeric value in the range between 0 (strongly disagree) and 5 (strongly agree). We have excluded the studies that received lowest scores (i.e. 0 or 1) in at least 2 quality assessment criteria. The final set of papers consisted of 22 studies; their detailed classification is given in Section 3. The prevailing number of relevant studies dates back to the years 2015 and 2017. Limitations of our study are dictated by the choice of the electronic publication databases, formulated search queries, and correspondence to IC, EC, and Q1-Q4.

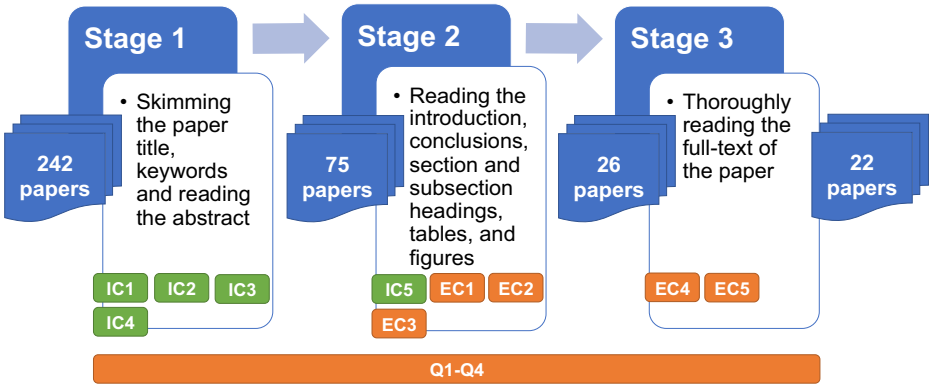

Figure 1. Paper selection stages and exclusion/inclusion/quality criteria.

2.2.4. Contents of the Studies at Glance

Before classifying the full-text of each study that got selected (see Section 3), we performed an analysis of all the 22 abstracts with an aim to see the most commonly used terms and the weighed ranking of the terms in the given set of abstracts. We designed and executed a process in RapidMiner[2] that included: tokenization, case transformation, filtering of English stopwords, the Porter stemming algorithm, and token filtering (minimum length = 3). We applied typical techniques for text data mining such as TF (*term frequency*) and TF-IDF (*term frequency - inverse document frequency*) to generate two word vectors respectively. Finally, we have calculated the average values of TF and TF-IDF for each of the remaining terms in the abstracts.

In Figure 2, we demonstrate the statistics on Top-20 most commonly used terms in post-processed abstract texts. Naturally, "data", "big", and "requir(ement)" take leading positions as these were the obligatory keyword in search queries. At the same time, less frequent terms are action-oriented (e.g. "analysi(s)", "process", "propos(e)", "gener(ate)", "develop") and could refer to the type of the contribution stated (e.g. "model", "applic(ation)", "techniqu(e)", "project", "approach", "servic(e)").

The purpose of TF-IDF is to penalize the most frequently used terms by reducing their weight as they are considered as noise. Figure 3 gives us a different Top-20 summary, where a difference in average TF-IDF values is not so dramatic, but the leading positions are taken by "servic(e)", "user", and "model". Implementation-related terms (e.g. "project", "analysi(s)", "applic(ation)", "techniqu(e)", "artefact", "develop(ment)", "software") outnumber the theory-related ones (e.g. "research", "formal", "scientif(ic)", "aspect"). We can also distinguish terms that indicate the study context (e.g. "genom(ics)", "manufactur(ing)", "system", "web", "domain").

[2] https://rapidminer.com/

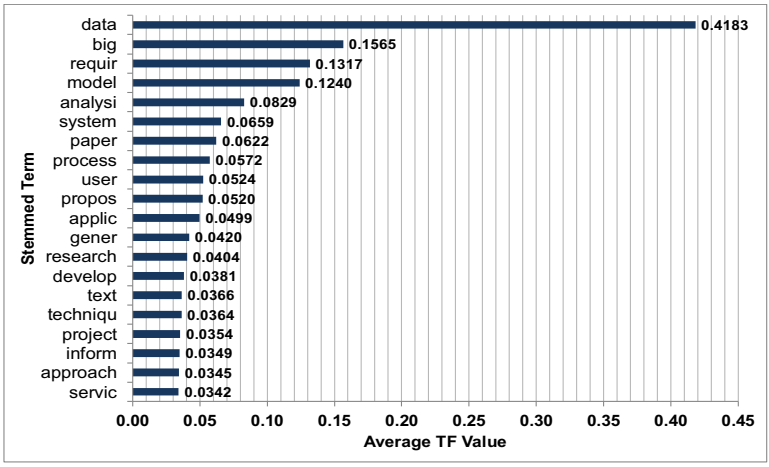

Figure 2. Stemmed terms and average TF values in the set of abstracts.

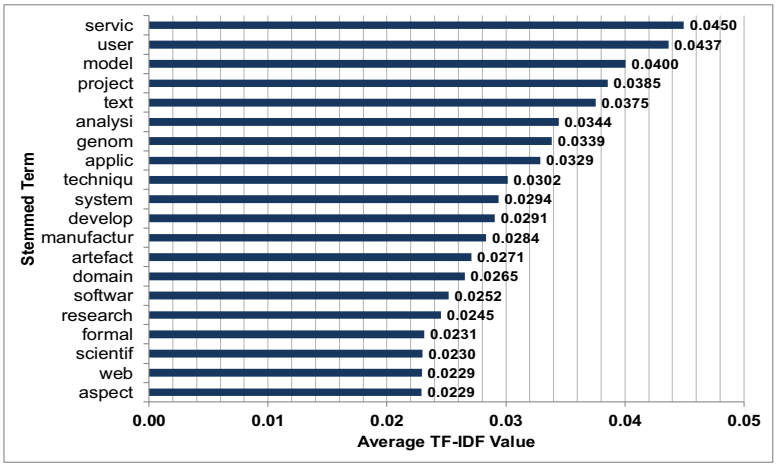

Figure 3. Stemmed terms and average TF-IDF values in the set of abstracts.

3. Classification of the Review Results

In this section we present an overview of the studies selected for an in-depth analysis. We summarize the main features of each paper in Tables 2-4. We elaborate on the results of our literature study by providing answers to the stated research questions.

3.1. Evaluation of the Selected Studies

Table 2 provides a full list of analyzed papers and their general features:

- Type of the contributions in the paper (can be more than one category): methodology, empirical study, case study, guidelines, etc.;
- Level of abstraction (*LoA*): high, medium (*med*), or low;
- Motivation for research: the main reason for conducting the study;

- Domain-independent (*D-I*): yes or no.

Table 2. General characteristics.

Study	Type of the contributions	LoA	Motivation for research	D-I
[7]	DLC model, use cases	high	the absence of a general DLC model to easily adapt to a different/new scenario	Yes
[8]	application scenarios	med	3 main challenges require consolidated solutions: data storage, data analytics, and data integration	No
[9]	guidelines (on structure of a Hive data model), empirical study	high	the lack of methodological guidelines for defining the structure of the tables in Hive	Yes
[10]	methodology (to create the model), application scenario	med	the absence of the RE artefact models to support RE process design and project understanding	Yes
[11]	case study	low	no research that examined GORE method/framework in Big data software development	No
[4]	methodology, case study	high	the lack of methods to manage, analyze, and visualize Big data systematically	Yes
[12]	methodology, survey	med	insufficient early involvement of users and stakeholders for better understanding of project goals	Yes
[13]	framework, case study	low	intensive disk I/O operations and focus on optimizing individual analysis tasks are the biggest bottleneck of existing genomics analysis pipelines	No
[14]	framework, empirical study	med	the absence of works to integrate phrase extraction and phrasal segmentation	Yes
[15]	methodology	high	improvement of the classical NLP techniques to be able work with Big data	Yes
[16]	methodology, empirical study	low	improvement of ETL process with mapReduce technologies	Yes
[17]	methodology	high	development of Big data as SOS technology	Yes
[18]	methodology	low	discovery of the IS processes with Semantic web technologies	Yes
[19]	case study (challenges and examples)	med	development of the context model for Big data software engineering	Yes
[20]	case study	med	companies don't use all available data to improve factoring	Yes
[21]	empirical study	low	development of the Ophidia framework to support (meta-)data management	Yes
[22]	methodology (requirement-driven approach for Big data application services)	med	the lack of requirement engineering guidelines to develop Big data services	Yes
[23]	methodology for the regeneration of the user interfaces for component based Web applications at runtime	med	necessity for component-based user interfaces to be intelligent and evolve over time	No
[24]	pattern-driven analysis requirements modeling method	high	difficulties to describe and understand the analysis problems due to their abstract nature	Yes
[25]	a tool for automated web mining and Big data analysis to generate training data for supervised architecture-traceability techniques	med	creating and maintaining traceability links between requirements, architecture, and source code is costly and complicated	Yes
[26]	architecture of a system	low	the necessity of execution of different NLP applications using parallel computing methods	Yes
[27]	method for building specifications of systems based on the data sets which offer system requirement information	high	rapidly growing data sets provide the possibility to extract useful information including the information about systems' requirements	No

Table 3 includes only studies that cover aspects of the requirement engineering, and are characterized according to the following criteria:

- Requirement artifacts: goals, scenarios, solution-oriented, etc.;
- Requirement development (*RD*) activities in focus (can be more than one category): elicitation, analysis, specification, validation, not specified (*N/A*);
- Requirement processing techniques: an outline of the proposed methods;
- (Semi-) Automatization capabilities (*S/A cap.*): high, medium (*med*), low, or not specified (*N/A*).

Table 3. Requirements-related aspects.

Study	Requirement artifacts	RD activities	Requirement processing techniques	S/A cap.
[7]	scenarios (applicability of the DLC in different projects)	all (no details)	direct/indirect data collection from sources, managing the ranges of sources, exploring and discovering new sources	N/A
[9]	solution-oriented (MD data model to tabular form)	elicitation, analysis	MD data models are constructed in compliance with analytical requirements defined by users	high
[10]	goals, scenarios (showing the capabilities of RE artefact model)	elicitation, specification	requirements, constraints, and scenarios gathered during interviews are grouped according to the classes in the artefact model	low
[11]	goals	elicitation, specification	requirements generation from goal-oriented models (i* and KAOS)	low
[4]	goals	all	queries are composed according to the IR; the source data is queried for availability; the raw data is transformed into facts and dimensions; MD data models of each data source are iteratively integrated into one	med
[12]	goals	elicitation, analysis	elicitation of the requirements on the BDA-as-a-service, construction of the Kano Questionnaire (KQ) for their classification, KQ analysis	med
[17]	requirements model	elicitation, analysis	modeling requirements from user needs and "BigData 7V"	low
[19]	scenarios	all	eliciting behavioral scenarios of the desirable system responses	low
[22]	solution-oriented (service requirements)	elicitation, analysis	defining new requirements based on experience and iterative implementation of them according to user and business needs	low
[23]	solution-oriented (for creating evolutionary mashup user interfaces)	all (no details)	capturing and storing user interaction data, applying ML algorithms to analyze interaction data, model transformation methods to get interface conversion rules	low
[24]	goals, analysis patterns	all (no details)	analysis patterns are used, analysis requirements are modeled	low
[25]	solution-oriented (requirements traceability support)	validation	architectural choices known as tactics are used, web mining methods, Big data analysis techniques	high
[27]	solution-oriented (requirements formal model)	specification	the data is restricted to the sequence of actions that the system behave, the models are the abstract automata	N/A

Table 4 includes interesting aspects from the field of Big data such as:

- Applicability of requirement development (*RD*) activities in Big data context: high, medium (*med*), low, not specified (*N/A*);
- Structured/Unstructured data processing capabilities (*S/U*): structured (*S*), unstructured (*U*), both, or not specified (*N/A*);
- Data processing techniques: an outline of the proposed methods if any;
- V-characteristics (*Vs*): (varies from 3Vs to 7Vs or not specified (*N/A*)).

Table 4. Big data related aspects.

Study	Applicability of RD activities in Big data context	S/U	Data processing techniques	Vs
[7]	high (fits the context of Big data well and is adjustable to any scenario to manage any kind of requirements keeping the high level of data quality)	both	transformations, quality check, pre-processing, post-processing	6Vs: Volume, Variety, Velocity, Value, Veracity, Variability
[9]	high (covers the aspects of integration of the multidimensional data sources into the Hadoop lifecycle)	U	automatic rule-based transformation of a MD data model into a Hive tabular schema	N/A
[10]	high (Big data requirements and scenarios are separate classes of the artefact model)	both	N/A	3Vs: Volume, Variety, Velocity
[11]	med (4 general requirements for Big data application were modeled as softgoals)	both	transformations, quality check	4Vs: Volume, Variety, Velocity, Veracity
[4]	high (covers management, analysis, and visualization of Big Data)	both	data stages definition, data sources acquisition and management, adding value to the data, implementation of a BDW, visualizations for Big Data	5Vs: Volume, Variety, Velocity, Value, Veracity
[12]	high (the framework includes customizable models of the Big Data Analytics process and its artifacts)	both	summarization and graphical representation of Kano questionnaires' results	N/A
[13]	N/A	U	genomics data acquisition and parsing, ETL processes, pre-processing, analysis, visualization	N/A
[14]	med (unstructured text can be transformed into structured units)	U	quality phrase mining, NLP	N/A
[15]	N/A	U	data mining, NLP	N/A
[16]	N/A	S	ETL, mapReduce	N/A
[17]	high (method describes Big data specific requirements)	N/A	system of systems (SoS)	7Vs: Volume, Velocity, Variety, Veracity, Value, Variability and Visualization
[18]	med (using the semantic technologies in Big data file system to discover IR)	S	semantic Web technologies	N/A
[19]	high	both	Multi-Peak, granular computing	4Vs: Velocity, Volume, Variety, Veracity
[20]	N/A	both	simulations	N/A
[21]	N/A	both	Big data technologies	N/A
[22]	high (Big data application service selection based on requirements catalog)	both	depend on selected services combined in the service pipelines for Big data processes	N/A
[23]	high (user interaction data processing with Big data technologies to get new requirements for user interface evolution)	both	data storage, view definition, transformations are performed by ML algorithms	N/A
[24]	high (no details specified)	N/A	only theoretical model is provided	N/A
[25]	high (dataset generation for traceability techniques from Big data sources)	both	document indexing techniques for indexing and searching	N/A
[26]	N/A	U	Map/Reduce, TF-IDF relevance function for keywords extraction	N/A
[27]	N/A	N/A	Big data as a data source to represent the systems' behavior and requirements	N/A

3.2. Evidence of the Requirement Development Activities in Big Data Projects

We have united the RQ1 and RQ2, since most of the corresponding papers cover both a theoretical approach and an empirical study. To analyze the evidence of the requirements development activities in Big data projects, we have taken the intersection of the studies included in Tables 3-4, which resulted in 13 papers. We grouped them by the column "Requirement development activities in focus" of the Table 3. In total, there are 5 groups: (i) all (elicitation, analysis, specification, and validation), (ii) elicitation and analysis, (iii) elicitation and specification, (iv) specification, and (v) validation. Let's consider the main contributions of papers that fall into each of the groups. These contributions are analyzed in detail in our paper [28].

3.2.1. Group (i): All Requirement Development Activities

The central figure of the study [7] is an advanced Data LifeCycle (DLC) model, which consists of 3 interconnected blocks: Data Acquisition, Data Processing, and Data Preservation.

Tardio et al. [4] present another methodology that includes 5 phases: (1) data stages definition, (2) data sources acquisition and management, (3) adding value to the data, (4) selection and implementation of a Big Data Warehouse (BDW) and, finally, (5) development of visualizations for Big Data. During stage (1) information requirements are identified. Stage (3) starts with the exploration of raw data sources. For each IR (from stage 1), the source data are queried to check its availability and potential usefulness. Then, the raw data will be converted into facts and dimensions.

In [19] a Big data software engineering contextual model is generated. The authors consider requirements engineering as one of the major Big Data challenges. Thus, they suggest creating domain models, eliciting use-case scenarios and requirements from stakeholders and other sources, developing functional and behavioral models, performing analysis, prioritization, and validation.

A method [23] that allows the component-based interfaces to be up-to-date with the changing user requirements is proposed. The requirements can change over time when new users or components appear. The methodology consists of 5 steps. The captured interaction data in (S1) is stored using Big data techniques. The goal of (S2) is to select appropriate data for different ML algorithms used in (S3) to extract new rules for interface improvements. (S4) transforms the outputs of ML algorithms in the conversion rules that can be applied to user interface. (S5) provides evaluation rules.

An analysis requirements modeling method [24] is based on analysis patterns reusing the previous experience to elicit and model analysis requirements by documenting the problem, goals, and analysis models. The description of an analysis pattern is based on a template. The method consists of 5 phases that allow eliciting analysis requirements.

3.2.2. Group (ii): Elicitation and Analysis

An advantage of the approach [9] is considered to be the absence of the necessity to specify IR after the multidimensional (MD) data model is defined. A set of rules to transform automatically an existing MD data model into a tabular schema that can be implemented in Hive and queried with HiveQL is proposed.

Ardagna et al. [12] involve users and stakeholders in the requirement elicitation phase with an objective to prioritize the requirements. An initial list of requirements is

created using a uniform specification format: Name/ Property/ Rationale/ Scope, Source, and Target/ Priority/ Dependencies.

Yasin et al. [22] focuses on requirements processing for Big data application services. The services computing in the Big data context means development of service pipelines to support processing of Big data volumes. The whole Big data lifecycle is covered. When appropriate, the approach suggests the usage of existing services that correspond to the business needs.

3.2.3. Group (iii): Elicitation and Specification

The recent study [10] sheds light on post-processing of elicited, analyzed, prioritized, and specified requirements, proposes a RE artefact model for Big data end-user applications, and a method to create it.

A generic requirement model [11] for Big Data application is proposed. While being supplemented with Big data related classes, the model mostly makes use of the existing i* Framework and Knowledge Acquisition autOmated Specification (KAOS).

Tikito & Souissi [17] claim that Big data should be treated as a System of Systems. The method forms a model from all the collected data in 3 stages: (1) identification of the requirements from user requirements and requirements related to the Big data constraints, (2) definition of criteria for each requirement, which serves as a basis for the information obtained in stage (1), and (3) modeling the requirements.

3.2.4. Group (iv): Specification

Zhang et al. [27] propose a software design method based on requirement data collected from users and describe the systems' behavior. The requirement data is analyzed in order to get formal specifications of systems. The authors consider only the type of data that are sequence of actions of the systems and the models are abstract automata.

3.2.5. Group (v): Validation

Santos et al. [25] describe techniques based on supervised machine learning algorithms, which allow tracing links between requirements, architecture, and source code. The authors do not provide new methods, but show how by means of automated tool usage that is based on existing web mining and Big data analysis methods problems with dataset generation for supervised architecture-traceability techniques can be solved.

3.3. Evidence of the (Semi-) Automatization Capabilities

Another objective of our study was to reveal possible (semi-) automatization of IR. To provide answers to the RQ3, we have selected the studies with "high" or "medium" values of the column "(Semi-) Automatization capabilities" in Table 3.

3.3.1. High-level (Semi-) Automatization

Santos & Costa [9] introduce a set of rules that automatically transform a MD data model into a tabular schema in Hive. First, the dimensional lattice that incorporates dimensions and all the combinations between dimensions is generated. Then, column groups (descriptive for attributes and analytical for measures or business indicators) are

detected. Next, another rules allow to associate column groups of both types to physical Hive tables, and define aggregation functions. Then, columns in Hive tables are classified either as descriptive or analytical depending on the column group they originate from. Lastly, another set of rules ensures partitioning and bucketing.

The goal of the BUDGET tool in [25] is dataset generation for traceability techniques; therefore, the tool has following features. (1) Open source systems code repository (from GitHub, SourceForge, Apache, and Google Code). (2) The web-mining component that uses Google Search API to search technical specifications of tactics in different technical libraries (e.g. MSDN). The authors define tactics as building blocks of the software architecture, which satisfy some quality requirements. (3) An automated Big-data analysis engine that uses repository from (1) to extract sample implementations of tactics. (4) The tool supports different data-sampling strategies (stratified and random sampling techniques). (5) Filtering feature allows to get more targeted results, e.g. in specific programming language.

3.3.2. Medium-level (Semi-) Automatization

Tardio et al. [4] automatically determine the availability and usefulness of the source data; sources are queried with tools like Apache Pig or Hive, while queries are formed in compliance with information requirements. Then, the raw data is translated into facts and dimensions according to the results of querying. Unfortunately, there is no suggestion in [4] on how to automate generation of queries to source data based on requirements, or how to distinguish IR stated for facts, measures, or dimensions. A semi-automated approach is applied at the integration stage: fact similarities of each MD data model are calculated, similar MD data models are grouped, a MD data model is created for each group, and finally, to integrate multiple models into one, remaining MD elements are compared in a non-automated way.

An approach for requirement prioritization put forward in [12] could be characterized as semi-automatic. It includes the composition of the Kano Questionnaire according to the list of elicited requirements. The Kano Questionnaire in fact is a survey that includes functional and dysfunctional questions with multiple-choice answers that are mapped to numeric values. For evaluation of each requirement, for example, assigning Functional (F), Dysfunctional (D), and Importance (I) scores can be applied that are calculated by corresponding formulae, while I is an average importance value.

3.4. Application of Natural Language Processing (NLP) Techniques

In this section, we give a brief overview of the methods that employ already existing data mining techniques with an aim to discover new information. In our opinion, NLP approaches mentioned in this section can be applied for generating potentially useful and previously unstated information requirements. However, one of the main challenges is to successfully integrate existing data mining techniques into the Big data ecosystem. More than that, in the course of our study we have found out that none of the approaches has been applied in a straightforward manner for requirements analysis. Nevertheless, we are convinced that certain techniques might be valuable at the stage of (semi-) automatic processing of the information requirements.

Application of NLP tools in the context of Big data has been recognized a as challenge by the research community due to the fact that such kind of data as query

logs, social media messages, free-style e-mail communication, etc. does not necessarily obey the strict language rules. Therefore, studies that cover more general data-driven approaches primarily based on the frequent pattern mining principle took place. For example, Liu et al. [14] present a framework the purpose of which is to extract quality phrases from text corpora integrated with phrasal segmentation, this way transforming unstructured text into structured units.

A phrase is defined as a sequence of words that appear contiguously in the text and serves as a whole semantic unit in certain context of the given documents. Its raw frequency is the total count of its occurrences. Although there is no universally accepted definition of phrase quality, the authors [14] quantify phrase quality based on certain criteria. Using a value between 0 and 1 to indicate the quality of each phrase, four requirements of a good phrase are specified:

- Popularity: a feature that can change its value over time or occasions when a sequence of words is considered as an inseparable semantic unit (e.g. out of two terms "data base" and "database" the latter one is more popular);
- Concordance: the collocation of tokens in such frequency that is significantly higher than what is expected due to chance is considered as a whole semantic unit (e.g. "powerful tea" vs "strong tea", the latter one is more common in English);
- Informativeness: a phrase is informative, if it is indicative of a specific topic;
- Completeness: a complete phrase should be interpreted as a whole semantic unit in certain context (e.g. "relational database system", "relational database", "database system").

The advantage of the framework [14] is that only limited training is required to produce generated phrases of the quality that is close enough to human judgment. The method is scalable: both computation time and required space grow linearly as corpus size increases. Efficiently and accurately extracting quality phrases is the main goal of the study. The full procedure of phrase mining is as follows. (1) Generate frequent phrase candidates according to popularity requirement. (2) Estimate phrase quality based on features about concordance and informativeness requirements. (3) Estimate rectified frequency via phrasal segmentation. (4) Add segmentation-based features derived from rectified frequency into the feature set of phrase quality classifier; repeat step (2) and (3). (5) Filter phrases with low rectified frequencies to satisfy the completeness requirement.

Cheptsov et al. [15] claim that traditional word processing techniques and tools are not capable of handling large amounts of data. Naturally, problems arise because of the size, structural complexity, and frequency of updates of the text sets. The approach [15] allows getting a deeper insight into the document contents when processing it with data mining techniques. The solution improves existing data mining techniques by the algorithms based on ontological domain modeling, NLP, and ML. Ontology is being crafted to simulate a particular domain, while data mining technologies ensure automatic data retrieval from (un)structured data. NLP identifies new terms (e.g., Persons), relations (e.g., role in the company), or definitions. Cheptsov et al. [15] describe 2 examples of word processing techniques that couldn't have been feasible to implement with the existing tools available. The main advantage of the proposed ontology-based reasoning algorithm is that the ability to automate to a certain extent the process of knowledge retrieval. Another advantage of this technique is its

parallelization capabilities that ensure high performance on large-scale Cloud computing infrastructures.

Nesi et al. [26] propose a system that allows execution of different NLP applications that is based on open source GATE APIs and implemented via MapReduce on a multi-node Hadoop cluster. The architecture and implementation of the system is validated by a specific design case for keywords and keyphrases extraction from unstructured text. The unstructured text is gathered from the web by crawling phase and is processed later by keyword/keyphrases extraction process. The main task of the application is the relevance estimation of keywords and keyphrases within the domains' corresponding document set done by computing TF-IDF relevance function. The future work according to authors is implementing the parallel processing of keywords/keyphrases extraction module in other environment e.g. Spark and improve the keywords/keyphrases extraction by semantic features.

We consider that NLP technologies mentioned in our findings are suitable to be integrated into elicitation of requirements and their post-processing.

4. Discussion

Having taken a thorough look at works of other authors in the direction of requirement analysis in Big data context, we have shaped the impression that there is a shortage of studies that would explicitly describe how to analyze Big data in order to obtain information requirements in an automated/semi-automated manner. Nevertheless, there are articles whose authors share the same point of view with us; for example, authors [4, 10, 11, and 17] state that information requirements should be defined before the actual development, or that information requirements can be found after reading the data into a Big data file system as in [9, 14, and 15].

Our suggestion is to get information requirements in two stages: 1) before the project development, and 2) after loading the data into a Big data file system. Prior to initiate the project development, information requirements should be obtained in a similar fashion as it is typically done in business intelligence solutions, which is as follows:

- Analyze organizational processes and process descriptions;
- Specify user needs by means of, for example, user stories;
- Determine information requirements by analyzing the organizational goals;
- Define the information requirements precisely and align them with appropriate KPI indicators and measurement principles.

After determining the information requirements, the development of a Big data solution can begin and data can be saved in the file system. Once data is loaded into a Big data file system, it can be analyzed in order to:

- Discover new information requirements;
- Find new relationships between entities;
- Analyze the data quality;
- Define data gaps;
- Analyze data integration capabilities between different data sources.

New information requirements can be defined using *data mining* techniques such as, for instance, tracking patterns, classification, association rules, outlier detection, or clustering. However, the open research problem is how to introduce these analysis

approaches into the Big data ecosystem keeping in mind the main characteristics of Big data (in particular, Volume, Velocity, Variety, Variability, and Veracity).

Taking advantage of the Big data approach for the development of information systems, loading all the data available within the organization into the Big data file system (e.g. HDFS) is a common practice. Our suggestion is to focus on analyzing data loaded into the Big data file system to discover new information requirements using data mining algorithms and NLP technologies. Our position is that there still may be hidden or unobvious relationships between data to be detected using automated processes, which are able to find these relationships.

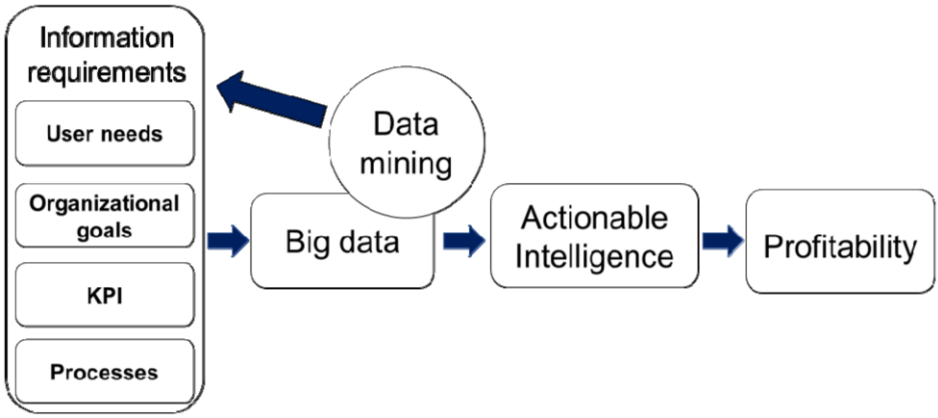

Figure 5. A schematic view of the information requirements definition and elicitation from Big data.

Figure 5 schematically demonstrates our vision of defining and obtaining information requirements in the Big data context (the diagram should be read from left to right). A new *information requirement* can be loaded into a *Big data* file system and analyzed afterwards. We anticipate that when designing a Big data solution, information requirements should be defined before the project development has started, identifying the needs of the end-users of the system to be developed (for instance, employing user stories), organizational goals, KPIs, and analyzing the processes in the organization. We believe that such approach to the analysis of information requirements is capable of supporting the daily activities of the organization, promoting growth and reflecting data, which in its turn, indicates that the organization is working in accordance with its objectives and strategy.

After one or more iterations that consists of defining information requirements, analyzing data, and discovering new requirements, employees of the organization are offered *actionable intelligence*, which means providing the information to customers who set the information requirements. There is a possibility that no information units can be delivered, because the data is not available in the organization, i.e. in its source systems. In such cases, we would recommend to review and improve the existing source information systems, and begin to accumulate the necessary data in the source systems, which in some cases may mean changing the current processes in the organization.

The last step in Figure 5 implies that the organization achieves *profitability* from the data provided. It is important to point out that the data delivered to the organization will be useful and will stimulate growth only if the information requirements were accurately defined during the first step. To be more precise, “accurately defined” IR

means that they are: a) in accordance with the strategy of the organization that reflects reality, (b) supporting the daily information needs of the system user, and (c) in line with the actual processes of the organization as well as will include planned changes.

5. Conclusions

The studies of the selected papers showed the different understanding of Big data starting from the most popular definition of Big data with 3Vs in [10]: *Volume*, *Variety*, and *Velocity*, followed by 4Vs in [11], [19] where *Veracity* was added to the list, 5Vs in [4] with *Value*, 6Vs in [7] with *Variability*, and, finally, 7Vs in [17] with *Visualization*. Moreover, 14 papers do not mention explicitly the definition of Big data used in the studies. One of the most important features that determines the methods used for the data processing is Variety. The analysis of the papers regarding the type of data variety showed that in most cases the discussed data processing capabilities were applicable to both Structured and Unstructured data, however, there were also specific methods only for processing structured data [16],[18] or unstructured data [9], [13], [14], [15], [26]. Some of the analyzed papers do not provide enough details, and, thus, the V-characteristics were classified as "not specified".

Summarizing the studies on Big data and requirements engineering, the importance of defining requirements while developing a Big data solution is indisputable. One of the reasons for Big data project failures is that not all the necessary information requirements are provided initially or the user's expectations are different (e.g. regarding data quality, data access, etc.). In some cases, the failure can be explained by the fact that no analytical steps had been carried out before the development.

We conclude that the requirements and data analysis should be integrated in 2 phases of development of the Big data solution. (1) Prior to the development it is possible to identify the user expectations timely and compare them with available resources, source data, quality, granularity, and available resources of the project i.e. budget, time, skills, and environment (e.g. as in [4, 10, 11, and 17]). (2) When the data has already been loaded from the source systems, a user may not be able to define all information requirements. However, when all the data is collected, new relationships and values can be explored (e.g. as in [9, 14, and 15]).

Acknowledgements

This work has been partially supported by University of Latvia project AAP2016/B032 "Innovative information technologies".

References

[1] M.A. Beyer and D. Laney, *The importance of 'big data': a definition*. Stamford, CT: Gartner, 2012.
[2] L. Kart, N. Heudecker, F. Buytendijk, *Survey Analysis: Big Data Adoption in 2013 Shows Substance Behind the Hype*. Gartner Inc., 2013.
[3] A. Katal, M. Wazid, R.H. Goudar, *Big data: issues, challenges, tools and good practices*. IC3'13, pp. 404-409, IEEE Press, 2013.

[4] R. Tardio, A. Mate, J. Trujillo, *An iterative methodology for big data management, analysis and visualization*. IEEE BigData 2015, pp. 545-550, 2015.
[5] F. Di Tria, E. Lefons, F. Tangorra, *Design process for Big Data Warehouses*. DSAA'14, pp. 512-518, 2014.
[6] B. Kitchenham, S. Charters, *Guidelines for performing systematic literature reviews in software engineering*. Technical report, Keele University, 2007.
[7] A. Sinaeepourfard, J. Garcia, X. Masip-Bruin, et al., *Towards a comprehensive data lifecycle model for big data environments*. BDCAT'16. ACM, New York, NY, USA, pp. 100-106, 2016.
[8] E.G. Caldarola, A. Picariello, D. Castelluccia, Modern Enterprises in the Bubble: Why Big Data Matters. *SIGSOFT Softw. Eng. Notes* **40**(1-2015), 1-4.
[9] M.Y. Santos and C. Costa, *Data Warehousing in Big Data: From Multidimensional to Tabular Data Models*. C3S2E '16. ACM, New York, NY, USA, pp. 51-60, 2016.
[10] D. Arruda and N.H. Madhavji, *Towards a requirements engineering artefact model in the context of big data software development projects: Research in progress*. IEEE Big Data 2017. pp. 2314-2319, 2017.
[11] H. Eridaputra, B. Hendradjaya, W.D. Sunindyo, *Modeling the requirements for big data application using goal oriented approach*. ICODSE'14, 2015.
[12] C. Ardagna, P. Ceravolo, G.L. Cota, et al., *What Are My Users Looking for When Preparing a Big Data Campaign*. IEEE BigData Congress 2017, pp. 201-208, 2017.
[13] T. Abdullah and A. Ahmet, *Genomics Analyser: A Big Data Framework for Analysing Genomics Data*. BDCAT'17. ACM, New York, NY, USA, pp. 189-197, 2017.
[14] J. Liu, J. Shang, C. Wang, et al., *Mining Quality Phrases from Massive Text Corpora*. SIGMOD'15. pp. 1729-1744, 2015.
[15] A. Cheptsov, et al. *Introducing a new scalable data-as-a-service cloud platform for enriching traditional text mining techniques by integrating ontology modelling and natural language processing*. WISE'13, Springer, Heidelberg, LNCS, vol. 8182, pp. 62-74, 2014.
[16] H. Mallek, et al., *BigDimETL: ETL for Multidimensional Big Data*. ISDA'16, Springer, Cham, AISC, vol. 557, pp. 935-944, 2016.
[17] I. Tikito and N. Souissi. *Data Collect Requirements Model*. BDCA'17. ACM, New York, NY, USA, Article 4, 7 pages, 2017.
[18] Ch. Di Francescomarino, et al., *Semantic-based process analysis*. International Semantic Web Conference. Springer, Cham, LNCS, vol. 8797, pp. 228-243, 2014.
[19] N.H. Madhavji, A. Miranskyy, K. Kontogiannis, *Big picture of big data software engineering: with example research challenges*. BIGDSE'15, IEEE Press, pp. 11-14, 2015.
[20] G. Shao, S. Shin, S. Jain, *Data analytics using simulation for smart manufacturing*. In: Proceedings of the Winter Simulation Conference, IEEE Press, pp. 2192-2203, 2014.
[21] S. Fiore, et al. *Big data analytics on large-scale scientific datasets in the INDIGO-datacloud project*. CF'15. ACM, New York, NY, USA, pp. 343-348, 2017.
[22] A. Yasin, et al., *Big Data Services Requirements Analysis*. APRES'17, Springer, Singapore, CCIS, vol. 809, pp. 3-14, 2017.
[23] A. Fernandez-Garcia, et al., *Evolving mashup interfaces using a distributed machine learning and model transformation methodology*. OTM'15 Workshops, Springer, Cham, pp. 401-410, 2015.
[24] J. Ji and R. Peng, *An analysis pattern driven requirements modeling method*. REW Workshops, IEEE International, IEEE Press, pp. 316-319, 2016.
[25] J.C. Santos, et al., *BUDGET: A Tool for Supporting Software Architecture Traceability Research*. WICSA'16, IEEE Press, pp. 303-306, 2016.
[26] P. Nesi, G. Pantaleo, G. Sanesi, A hadoop based platform for natural language processing of web pages and documents. *Journal of Visual Languages & Computing* **31** (2015), 130-138.
[27] Y. Zhang, Y. Chen, Y. Ma, *A Framework for Data-Driven Automata Design*. In: Requirements Engineering in the Big Data Era. Springer, Berlin, Heidelberg, CCIS, vol. 558, pp. 33-47, 2015.
[28] N. Kozmina, L. Niedrite, J. Zemnickis, *Information Requirements for Big Data Projects: A Review of State-of-the-Art Approaches*. In: Lupeikiene A., Vasilecas O., Dzemyda G. (eds) DB&IS 2018. Springer, Cham, CCIS, vol. 838, pp. 73-89, 2018.