DataBench

Evidence Based Big Data Benchmarking to Improve Business Performance

D5.1 Initial Evaluation of DataBench Metrics

Abstract

This deliverable is about setting-up an ontological and analytic infrastructure to perform technical evaluation of DataBench metrics collected in WP1-WP2. The result of this deliverable is to prepare the plan on how the technical validation will be structured, measured and reported. The tangible part of the deliverable is the DataBench ontology to be used in the DataBench Toolbox.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 780966

Deliverable D5.1	Initial Evaluation of DataBench Metrics
Work package	5
Task	5.1
Due date	31/12/2018
Submission date	22/11/2019
Deliverable lead	JSI
Version	2.0
Authors	JSI (Marko Grobelnik, Maja Skrjanc)
Reviewers	Todor Ivanov (GUF), Tomas Pariente Lobo (ATOS), Arne Berre (SINTEF)

Keywords

Big Data Validation, Ontology Design, Knowledge Graphs, Meta Learning

Disclaimer

Copyright Notice

Table of Contents

Table of Figures

Executive Summary

This deliverable describes the preparation of data and an analytic infrastructure to perform technical evaluation of DataBench metrics collected across workpackages 1-4. The deliverable details how the technical validation will be structured, measured and reported.

The tangible part of the results is the DataBench ontology structuring the domain of benchmarking along several dimensions and Big Data/machine learning experimentation.

The key challenge in Big Data benchmarking is dealing with diversity of decisions that a stakeholder in a data processing related activity (like data engineers, data analyst, decision makers) can make at the different stages in the analytics pipeline process. The key stages include data storage, data preparation, model construction, data and model representation, analytics, and model interpretation.

In our approach, the plan is:

a) to structure the domain of Big Data process along many indicators in a form of a Knowledge Graph based on an upper ontological structure;
b) to make benchmarking data comparable from different benchmarking applications run on the DataBench tool;
c) to use the collected knowledge usable for daily tasks of data scientists end users in a form of recommendation service or interpretable aggregated knowledge.

WP5 will closely collaborate and provide technical input to the DataBench Toolbox (WP3) and provide a baseline for the evaluation exercise in WP4.

1. Introduction

Technology benchmarking focuses on comparing different technological solutions and building blocks along different dimensions, measured via empirically observed indicators. Ideally, the measured environment is controllable to perform pivoting of different system parameters and to compare the outcomes. In the case of Big Data, the benchmarking approaches follow the same philosophy – however, an important issue is the complexity of an average Big Data project with many tunable parameters along the stages of a typical pipeline. Many of these parameters can significantly change the overall performance of the executed benchmark.

Since the problem is not easy [1] and cannot be entirely solved in a clear rigorous scientific manner, we will approach the problem of an overall evaluation and validation of the collected metrics in a practical way to produce a result satisfying end-user needs.

The approach consists from the two major steps: (1) constructing the DataBench ontology by structuring all the inputs from within the project (WP1-WP4), and (2) including relevant external resources (like MLSchema [10] and DCAT [11] ontologies) to complement DataBench specifics.

In particular, the DataBench ontology is taking numerous inputs from D1.1, D1.2 and D1.4 where the domain of benchmarking is described in a more qualitative form with underlining a number of indicators along several dimensions (business, application, platform & architecture, and benchmark specific/ecosystem).

The purpose for the ontology is to represent the actual benchmarking experimentation data in WP3 (D3.1) as part of the DataBench Toolbox. The ontology (as a conceptual structure) along with the Knowledge Graph (as concrete data alignes with the ontology) represent the data basis to collect and to analyse the benchmarking data. The DataBench Toolbox will use the structure and Knowledge Graph to acquire, store and reason with the collected data.

By our best knowledge, the DataBench ontology is the first trial to structure the domain of Big Data and broader data analytic Benchmarking by connecting it to other related ontological resources such as Machine Learning Schema (MLSchema) and Data Catalog (DCAT). Since the domain of Benchmarking is active and evolves, the plan is also to extend and evolve the DataBench ontology. In particular, in the next release of the DataBench ontology it will be extended with the upcoming ISO SC42 AI standard as one of the industrial baseline vocabularies for the area.

In the next sections we will first present the overall approach in the WP5 along the three tasks, and in the continuation, we will focus on the technical approach of Task 5.1.

2. WP5 Task Structure and Objectives

WP5 is expected to validate and assess the correspondence of the technical KPIs and metrics and the resulting benchmarks collected and refined in WP1-WP2 and integrated in the Toolbox developed in WP3, to make sure they effectively correspond to the intentions of the original tools and needs of industrial and research communities. The actual evaluation in WP4 will be served as a set of analytical insights into the data on different levels.

The work package consists of three tasks connected into a pipeline. Figures 1 and 2 depict the role of each of the following three tasks:

- **T5.1** – Systematization/ontologizing, storage, evaluation and validation of metrics based on data measurement including business data, sensor data/time series, media data, natural language data (incl. web/social media) and geospatial/spatiotemporal data;
    - *Objective: To structure description of benchmarking experiments through indicators. To create a Knowledge Graph and an ontology of indicators [Figure 3].*
- **T5.2** – Assessment of technical usability, scalability, complexity and relevance of corresponding metrics and data, considering problems to be solved;
    - *Objective: To generalize data from benchmarking experiments into a model via machine learning and other analytical techniques on the collected data from WP1 & WP2.*
- **T5.3** – Assessment of the sustainability of the tool, finalisation of the methodology;
    - *Objective: Integration of the tool into the DataBench Toolbox and positioning the DataBench methodology to be used in the data science industry.*

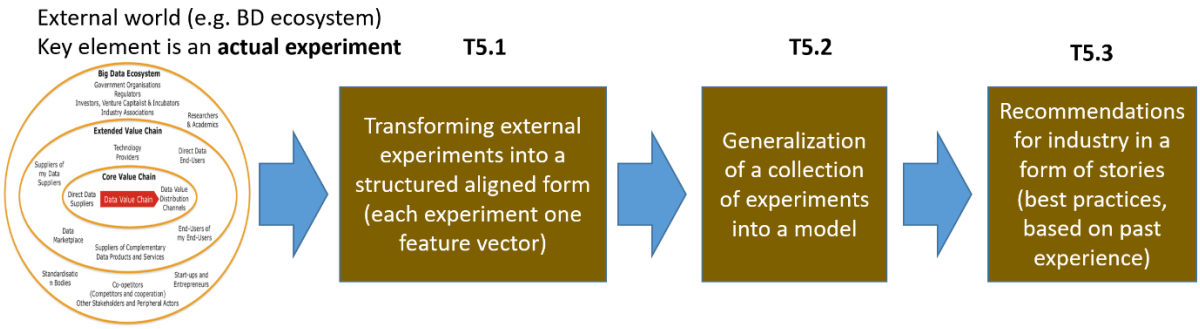

Figure 1 - The Pipeline of three Tasks in WP5

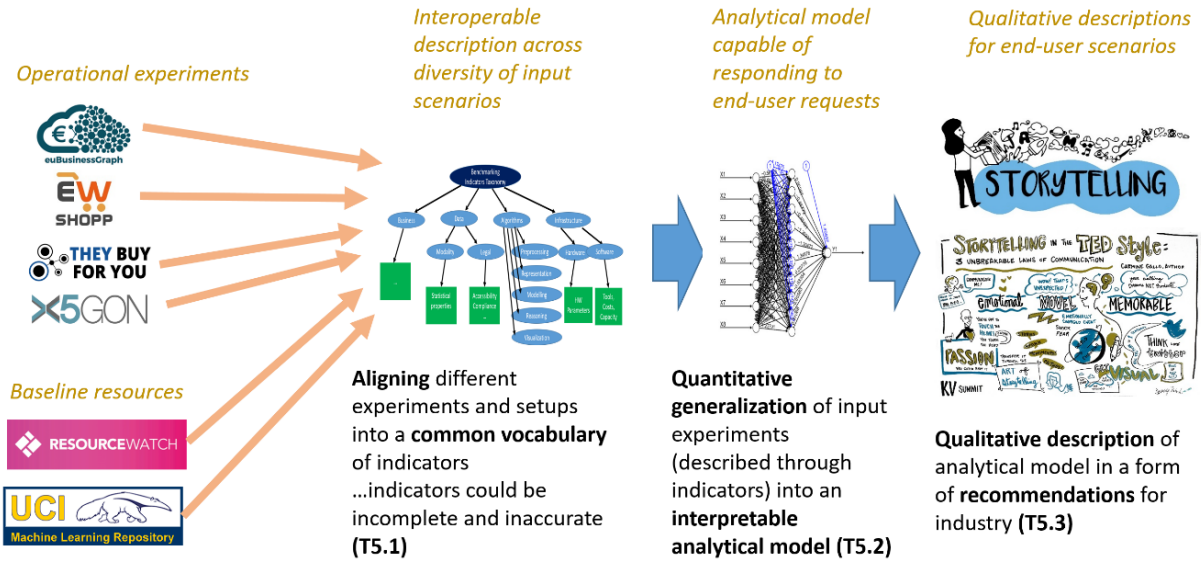

Figure 2 - Detailed Structure of WP5 along three Tasks

In other words, WP5 will make sure to:

- collect the data and corresponding data schemas from WP1/WP2 and external ones, and organize them into the DataBench ontology;
- collect requirements from WP3 (DataBench Toolbox) for the solution to get integrated on the software engineering level (most probably the solution will be REST API interface);
- collect requirements from WP4 on the required analytic (quantitative and qualitative) input;
- organize the data in a Knowledge Graph aligned with DataBench ontology;
- prepare analytic environment to allow extracting insights from the collected data from WP1 and WP2;
- prepare an analytical solution using collected data;
- integrate analytical solution into the DataBench Toolbox (WP3);
- support evaluation process of the WP4;
- finally, the end task of WP5 is to ensure sustainable deployment of the analytical solution to allow reuse of the environment beyond the end of the project.

3. Knowledge Graph Representation of DataBench Data (DataBenchKG)

Knowledge Graphs became around 2010 a predominant solution to store and retrieve structured data in environments which require flexible and ever changing schema. Historically, the area of Knowledge Graphs stems from the area of Semantic Web, where EU in its FP5, FP6, FP7, and H2020 programs invested large amount of resources, and as a consequence EU has visible role in the corresponding communities and delivers some of the core resources in the area.

In particular, Knowledge Graphs became the main solution to ensure interoperability in mid-sized to large enterprises, where the key role is to connect legacy data resources across the enterprises' IT infrastructure. Typically, Knowledge Graphs provide a thin infrastructure layer on the top of existing databases, connecting diverse data schemas and enabling data retrieval for flexible application scenarios.

As one of the key global resources in the area of Knowledge Graphs is "Linked Open Data Cloud" [3], which connects 1,234 datasets and schemas (as of June 2018). The LOD Cloud is maintained at Insight Centre for Data Analytics in Ireland with many international contributions. Figure 3 depics the structure of the LOD cloud.

How the Knowledge Graphs relate to the DataBench project? The information collected at various stages of the project (in particular WP1 and WP2) will be first organised in a structured form (DataBench ontology) to be easily accessible, structured along appropriate schemas, and interoperable with other related external semantic knowledge resources trying to standardize the domain of data management.

For that purpose, we will refer to the specific Knowledge Graph built in the DataBench project with the working name 'DataBenchKG'. In the next paragraphs we will describe the

key ingredients of DataBenchKG, the envisioned implementation and the required characteristics.
Based on the preliminary analysis, we envision the information coming from the project to include the following sources (but not limited to, in the case of necessity to expand):

- **Questionnaires** – question-answer pairs, where the question part will be textual, while the answer part will be either structured in the multiple-choice lists or in minority cases textual descriptions.
- **Interviews** – the data will include pairs of (semi)structured questions and answers as unstructured textual descriptions.
- **Data science algorithms descriptions** – algorithms will be described in a form of structured descriptions as used in data science; whenever possible, the descriptions will be aligned with an ontology of machine learning and broader data science related algorithms; as much as possible we will use the existing efforts as part of the W3C Machine Learning Schema[1].
- **Data science tools descriptions** – the tools (typically software systems) will be described in a form of structured descriptions; since we are not aware of any ontology to describe the tools and software packages, we plan to develop within the project a 'minimal viable ontology' satisfying the project needs.
- **Dataset descriptions** – to describe data, datasets and other types of data resources, we plan to use the existing body of knowledge from the area of Meta Learning (as a subfield of Machine Learning) [4][5]. The aim is to derive structured descriptions of characteristics of datasets which are commonly used in data science, statistics and broader in the area of data analytics. The process to extract data characteristics will be automated and will address general information about the datasets (modality, size) as well as shallow statistical properties (such as statistical distributions, correlation among the variables etc). During the course of the project we plan to compile a viable approach to define a schema satisfying the needs of the project. The major objective will be to automate the process of extracting such characteristics from diverse datasets.
- **Benchmarking tools description** – as part of the project we will address several benchmarking approaches and tools, and the goal will be to describe them in a structured way to perform comparison with particular focus on the DataBench platform. In particular, the aim is to connect in the DataBench Framework related initiatives, such as other H2020 projects and other experimental setups.
- **Benchmarking experiments** – each benchmarking experiment performed either within DataBench Framework or outside will measure and collect diverse KPIs (including memory consumption, time complexity, various metrics to estimate quality of analytic results, business aspects) – such collected data will be stored in the DataBenchKG in a structured way.
- **Benchmarking with machine learning models and datasets** – as part of the project we will perform extensive analytical tests combining a selection of ML and Big Data algorithms with a selection of publicly available reference data sets. The aim is to create a recommendation tool to suggest what kind of algorithms should be used in particular data scenarios. The purpose of this task is to derive a generalized understanding on how different machine learning and analytical algorithms and

[1] https://www.w3.org/community/ml-schema/

tools perform when analysing different datasets and under broad range of parametrizations. The models will be represented in a structured form where a possible common method to represent machine learning models in an interoperable way is 'Predictive Model Markup Language' / PMML[2].

The above types of information will be systemized and stored as a Knowledge Graph (DataBenchKG), where individual knowledge and data and fragments will be aligned with external ontologies/schemas and the ontology/schema constructed within the project (i.e. DataBench ontology). For technical and business concepts, where pre-existing semantic resources exists, we will align with the corresponding semantic schemas, like W3C Machine Learning Schema (describing machine learning experimentation), DCAT ontology (describing data resources) and Predictive Model Markup Language/PMML (describing machine learning models).

The collected data (aligned to DataBench ontology) will be stored conceptually as a Knowledge Graph, whereas for the implementation of the actual storage will plan to use one of the well established and proven scalable graph databases such as Neo4J (https://neo4j.com/), ArangoDB (https://www.arangodb.com/) or similar. The final decision, which graph database to be used for DataBenchKG, will be taken at the beginning of the implementation phase.

The aim of constructing DataBenchKG, is the aggregation and analytics capabilities on top of the collected data. Most of the data types and sources (as listed above), to be stored in the DataBenchKG, are of a moderate scale and consequently we do not expect major implementation issues. For these data resources the out-of-box graph database engine will support basic aggregation operations such as search and baseline statistics. As we plan, the data source with the most data input will be coming from the 'Benchmarking with machine learning models and datasets' (generated by the tools from WP5), where we expect tens of thousands (or more) experiments to be performed and stored in the graph data engine. For the purpose to be scalable and having ability to perform modelling and aggregation, we could use alternative data storage engines, like the NoSQL database MongoDB or relational database PostgreSQL. We see the structure of the graph of interconnected knowledge-graphs is color-coded addressing eight domains and cross-domain Knowledge Graphs in figure 3.

[2] https://en.wikipedia.org/wiki/Predictive_Model_Markup_Language

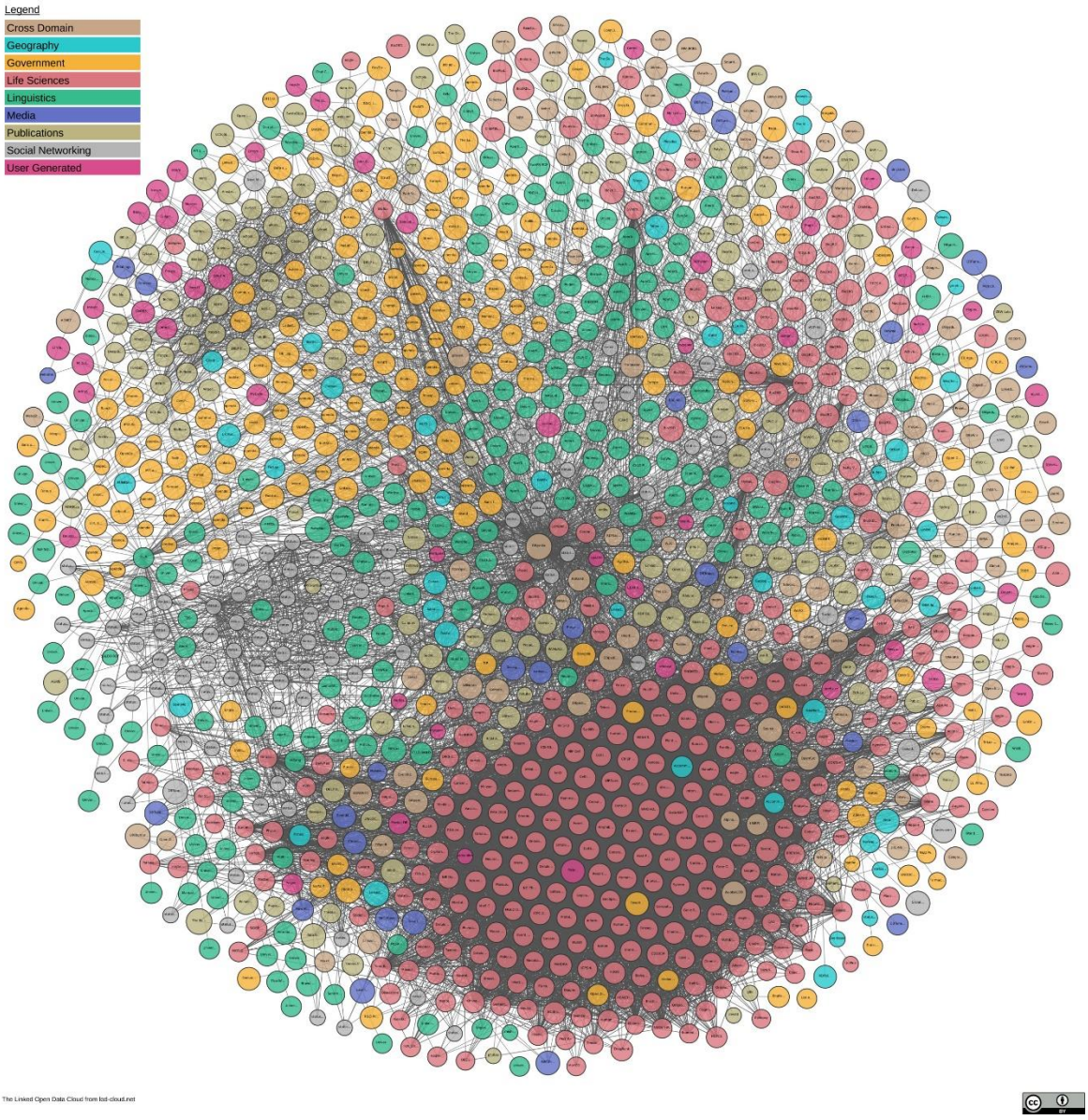

Figure 3 - Depiction of Linked Open Data Cloud [3]

4. Construction of the Ontology of Indicators/KPIs

4.1 Introduction

The previous section described major data sources which are planned to be collected and used in the DataBench project. The data will be stored as a Knowledge Graph, which is conceptually a lower level data structure, operating with schemas and allowing baseline operations such as retrieval, counting and simple statistics. The upper part of the structure, systemizing the domain of data science and consequently its benchmarking will be defined as an ontology with all the relevant concepts connected with a relationship structure.

As a short introduction to ontologies, we could say they consists of four major elements:

- Concepts which describe the anchors and fundamental part of the structure – typically they are represented as nodes in a graph/network structure.
- Relations which connect concepts into a structure and establish interplay between individual meaningful building blocks. Relations could be either hierarchical (which allows multi-level aggregate representation of the world), as well as horizontal (to represent data instances of the world to be described by an ontology). Relations are typically edges in the graph/network structure.
- Attributes are an additional element which further describes either concepts or in more rare cases relationships. Attributes are typically just fields with values to further elaborate the context of given ontological element.
- Data or Knowledge Sets are just data which need to be systemized into an ontology. These are typically data instances as measured and observed from the environment. In the case of DataBench, the Knowledge Graph will have this role, where all the data will be stored and inserted into the ontology with the purpose to generate complex queries and aggregation.

Figure 4 illustrates the major elements of an ontology.

The most popular formalism, standardized at W3C, is Web Ontology Language (OWL) [7]. It allows description of an ontology with all the above mentioned elements. Several tools allow input and manipulation with OWL ontologies – the most popular is Protégé [8] which will be used also in the DataBench project to define and manipulate the ontological structure.

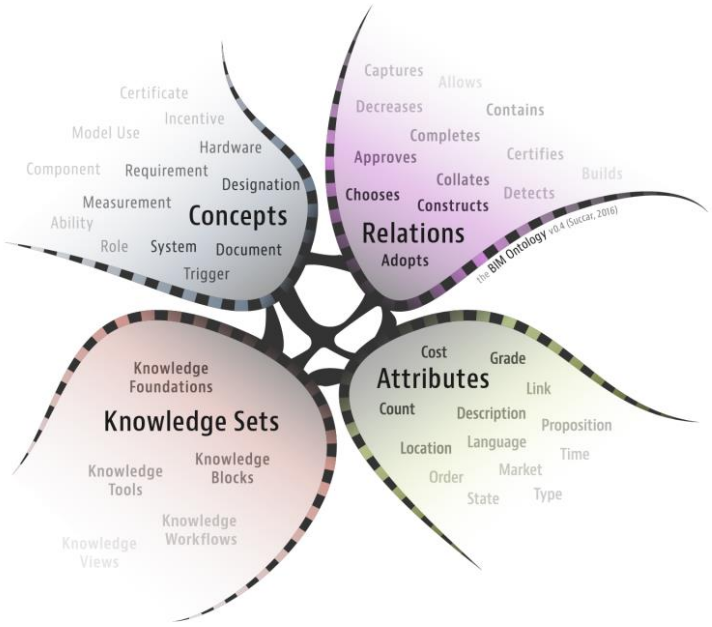

Figure 4 - Depiction of major Elements of an Ontology [6]

The DataBench ontology will use the standard approach of modelling concepts and relationships. The input will come from the data sources, as described in section 3. The major building blocks of the ontology are depicted on Figure 5 visualizing all the major data inputs within the project.

The key data types of the DataBench ontology will include raw data and collected (either calculated or estimated) KPIs. The major segments of the ontology will include:

- Business data and KPIs;
- Datasets including modality, characteristics and legal framework of the data accessibility;
- Algorithms including all the major subgroups like pre-processing, model representation, modelling/analytics process with associated parameters, model interpretation/reasoning and visualization;
- Infrastructure describing benchmarking setups like hardware and software with associated tools using the infrastructure;
- Information on the process pipeline like data acquisition (including Edge, IoT…), storage, pre-processing, curation, visualization, usage;
- Architectural approaches where benchmarks are executed.

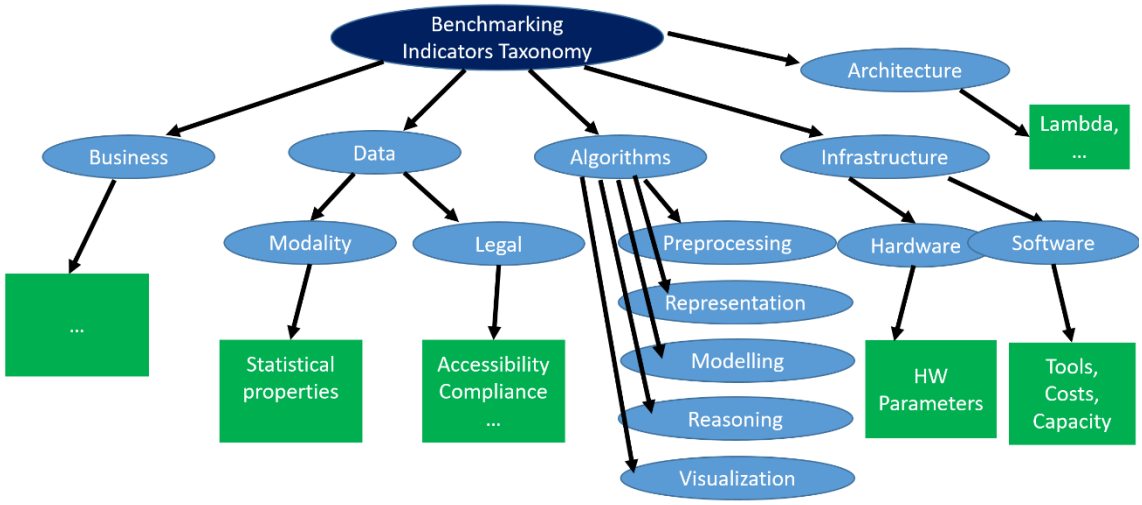

Figure 5 - Depiction of an Ontology of Indicators

The ontology structure (concepts relations, attributes) will be specified in collaboration with WP1, WP2 and WP3 which will contribute the data input to be organized. The operators to insert, aggregate and retrieve information from the ontology (either through lookup or some form of reasoning) will be defined in collaboration with WP3 (Toolbox) and WP4 (evaluation interpretation).

4.2 DataBench Ontology Design

DataBench ontology, in technical terms, is aimed to serve representation of Big Data benchmarking domain. Consequently, this requires representation of the key ingredients of a benchmarking process (including elements like experimenting and indicators) as well as benchmarking ecosystem (including elements like benchmarking organizations, industrial benchmarks and similar). To the best of our knowledge, there is no ontology of such kind developed and available – there are separate ontologies describing parts of the benchmarking domain, which we plan to reuse and integrate.

DataBench ontology will consist from the following modules:

- Benchmarking module – this module covers the domain of benchmarking in a broad sense with special focus on Big Data domain and even more specifically, the needs of the DataBench projects (as described in early WP1-WP4 deliverables). On the technical side this module covers the hierarchy of indicators to measure individual bechmarking experiments (covering from technical to business aspects). On the contextual part, this module describes a broader data analytic (Big Data, Machine Learning, Artificial Intelligence) benchmarking domain including organizations, industrial standards etc.
- Technical experimentation module – for this module we will reuse and extend the MLSchema ontology (http://www.w3.org/2016/10/mls/) [10] describing the elements and the process of machine learning experimentation. Since the area of machine learning process is in many ways a subset of the Big Data process, we will use MLSchema as a basis and extend it with infrastructural elements which put data analytics experimentation in a proper industrial context.

- Data module – this module describes the data part of benchmarking, as one of the core elements when dealing with data analytic experimentation. As basis for data description we will use DCAT ontology (https://www.w3.org/TR/vocab-dcat-2/) which extensively covers the meta data and data provenance aspects of data used in many possible contexts. Despite the breadth of DCAT, there are still technical process elements of data which are covered by the MLSchema.

The above approach and conceptual design might get extended in the future in the direction of detailing parts of the ontological structure.

In the following two sections we will present more in detail the three key modules of the DataBench ontology. The section will be finalized with the description of the DataBench ontology as the whole.

4.2.1 MLSchema Ontology Module

MLSchema (available from http://www.w3.org/2016/10/mls/) [10] was finalized in 2016 by a team of machine learning and semantic web experts to fill the missing gap of describing machine learning experimentation process. The work was performed as part of W3C standardization efforts. The actual specification in OWL format is available from <https://raw.githubusercontent.com/ML-Schema/core/master/MLSchema.ttl>

MLSchema is a light-weight ontology describing the overall machine learning process including experimentation, data, algorithmic process and generated models. It doesn't go very deep into details of each of the subparts (e.g. listing algorithms and corresponding parameters or evaluation techniques of specific data characteristics). For the purpose of the DataBench project individual classes are extended and more detailed. Since the area of machine learning is evolving, it is to be expected more things will be necessary to add to the subparts of MLSchema to be up-to-date and actionable.

Figure 6 depicts high level of the MLSchema core vocabulary. Specific classes, like "Algorithm", "Evaluation Procedure" or "Dataset Characteristics" need further elaboration to become useful in specific machine learning or Big Data scenarios. For this purpose we extended the relevant classes with more detailed specification. As an example, we show the specification of the "Algorithm" concept where we categorized machine learning algorithms (typically used in Big Data scenarios) in subtypes, and further each type in particular algorithm. Figure 7 depicts part of the "Algorithm" class categorization.

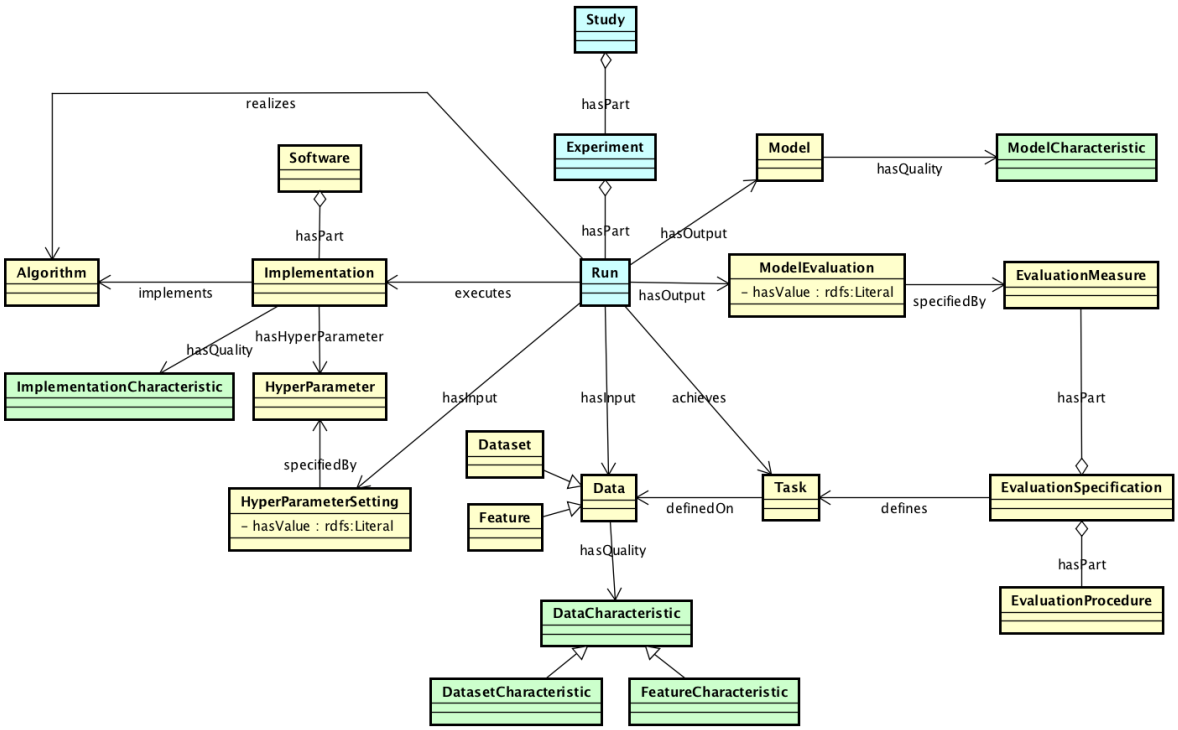

Figure 6 - The ML Schema core vocabulary.

Figure 6 depicts Information Entities as yellow boxes, Processes as blue boxes, and Qualities as green boxes.

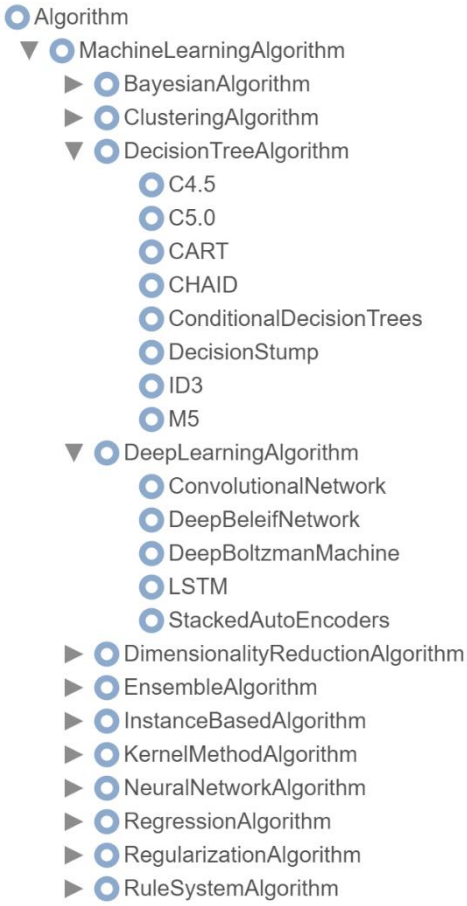

Figure 7 - The extended ML Schema "Algorithm" class

The extended ML Schema "Algorithm" class shown in figure 7 categorizes machine learning algorithms and the corresponding concrete instances of the algorithms.

Another, relevant to mention, DataBench extension of the MLSchema ontology is the concept of the "Hardware" where particular benchmarking experiment is running. This includes different types of hardware architectures, typical for the area of data analytics. Figure 8 depicts part of the DataBench ontology showing the experimentation setting.

The architectural aspect of the ontology could be expanded in many more details and requires very specific technical details of e.g. CPUs or GPUs being used in experimentation and other hardware elements like memory, data storage and connectivity if benchmarking would want to be specified in a very detailed way. Therefore, many standardized benchmarks fix parts of the architectural aspects to controlled and repeatable scenarios to test only e.g. algorithmic aspects of the experimentation. Within DataBench project we plan to stay on the level abstract enough to get into overly complex (and possibly inaccurate) descriptions of the computer architecture.

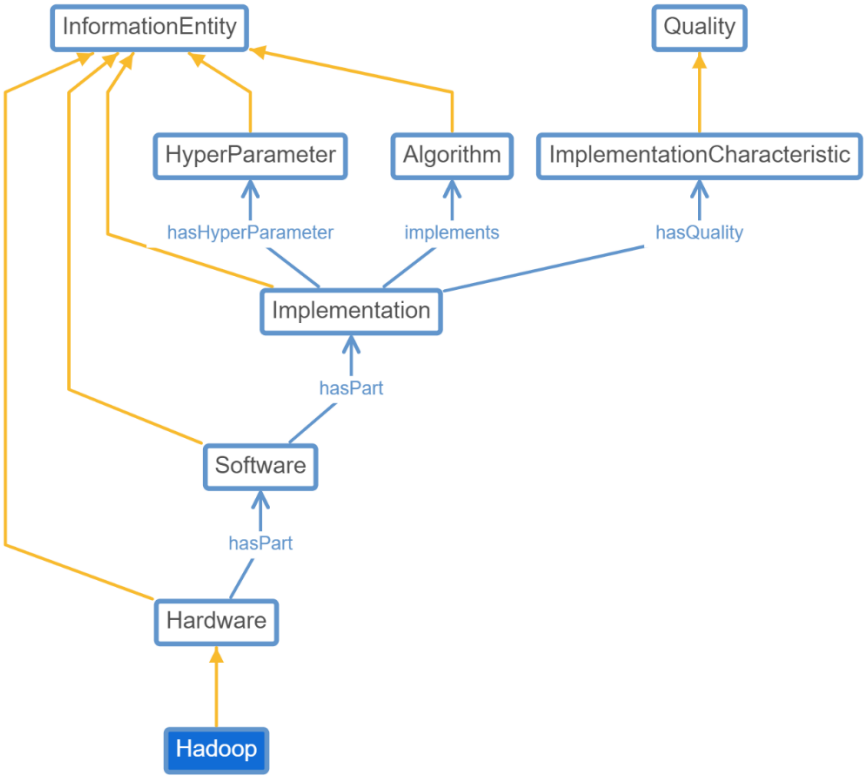

Figure 8 - MLSchema from "Hardware" to the "Algorithm" concept.

Figure 8 - shows the extended MLSchema ontology spanning from the "Hardware" concept to the "Algorithm" concept.

4.2.2 DCAT ontology module

DCAT ontology [11] is designed to facilitate interoperability between data catalogs published on the Web. It enables a publisher to describe datasets and data services in a catalog using a standard model and vocabulary that facilitates the consumption and aggregation of metadata from multiple catalogs. The general aim is to increase the discoverability of datasets and data services. Aggregated DCAT metadata can serve as a manifest file as part of the digital preservation process. The actual definition is available from <https://w3c.github.io/dxwg/dcat/rdf/dcat.ttl>.

The structure of DCAT consists from the following main submodules (summarized from [11]):

- **dcat:Catalog** represents a catalog, which is a dataset in which each individual item is a metadata record describing some resource; the scope of dcat:Catalog is collections of metadata about datasets or data services.
- **dcat:Resource** represents a dataset, a data service or any other resource that may be described by a metadata record in a catalog.
- **dcat:Dataset** represents a dataset. A dataset is a collection of data, published or curated by a single agent. Data comes in many forms including numbers, words, pixels, imagery, sound and other multi-media, and potentially other types, any of which might be collected into a dataset.
- **dcat:Distribution** represents an accessible form of a dataset such as a downloadable file.

- **dcat:DataService** represents a data service. A data service is a collection of operations accessible through an interface (API) that provide access to one or more datasets or data processing functions.
- **dcat:CatalogRecord** represents a metadata item in the catalog, primarily concerning the registration information, such as who added the item and when.

Figure 9 depicts the schema of the DCAT ontology with its main modules.

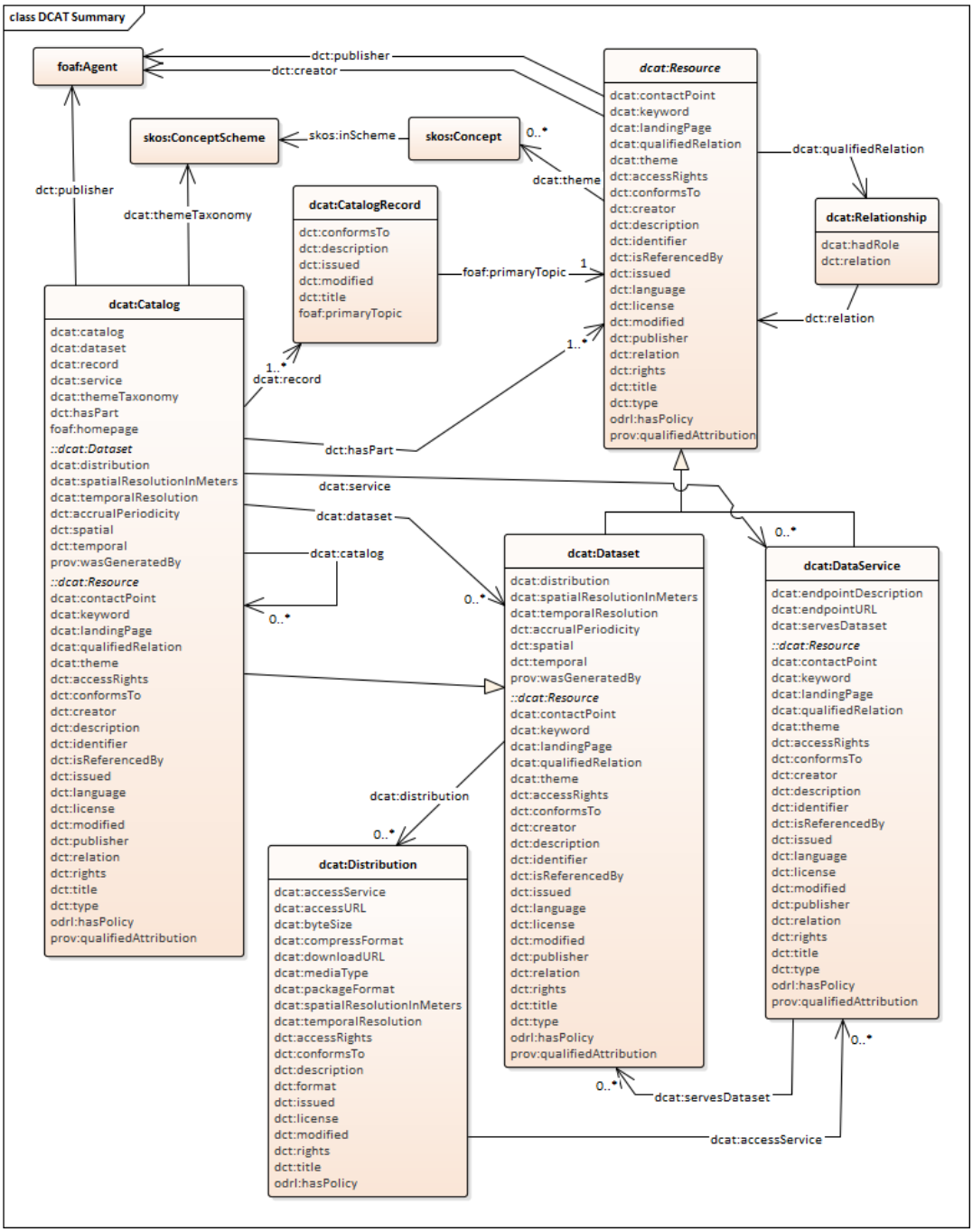

Figure 9 - Top level DCAT data model

Figure 9 - shows the classes of resources that can be members of a Catalog.

In the DataBench project the DCAT ontology is connected to several parts (shown in Figure 10): to the part of MLSchema module describing the data elements of the experimentation (in particular DataQuality) and to the Benchmarking module describing data specific indicators to be extracted and observed. While DCAT is very detailed in its meta-data description of data catalogs, within DataBench we plan to use only the main segments for the purpose of being practical. Namely, many practical scenarios don't include many of the segments covered by DCAT.

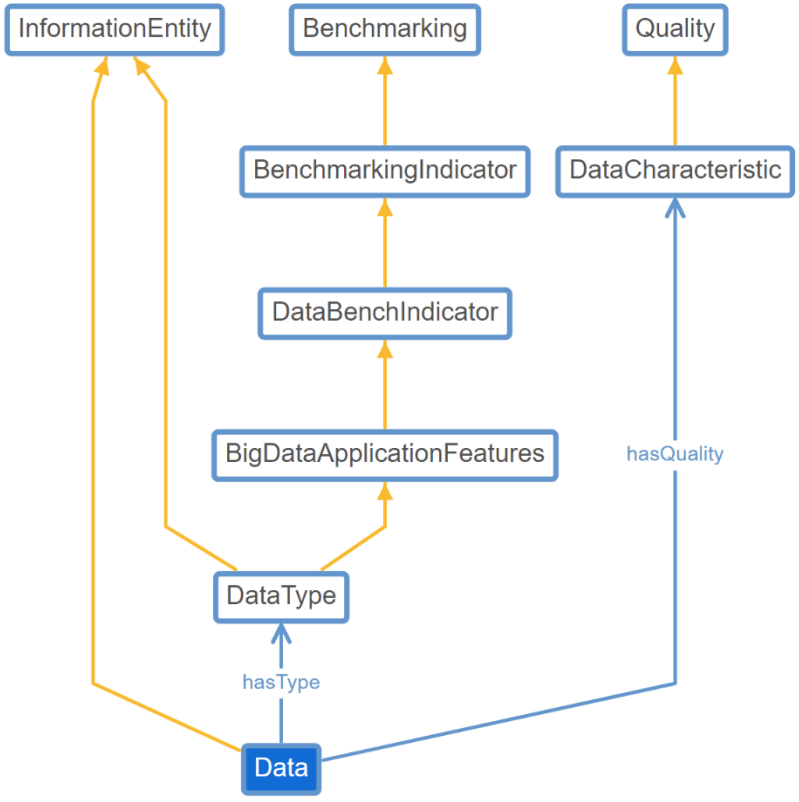

Figure 10 - Connecting points of the DCAT ontology into the DataBench ontology

4.2.3 Benchmarking Ontology Module

The benchmarking part of the DataBench ontology is the core module related to the DataBench project. The key aim is structuring the domain of benchmarking with a particular specialization in DataBench indicators developed in WP1-WP4 deliverables. In particular, D1.1 and D1.2 contributed most of the indicator information to the ontology, while D3.1 and D4.1 provided additional requirements and structure.

The benchmarking module has 4 key parts: Indicators, Domain, Organisation and Use Cases. Each of the parts has further elaboration with an aim to describe particular concrete benchmarks as well as benchmarking ecosystem of organizations, standards and use cases. The domain of benchmarking is spanning across many fields of science and research with many common characteristics but with lots of specifics which need to be detailed as per need basis. In the current implementation we have a strong focus on the related areas of Big

Data, Data Science, Machine Learning and data analytic part of AI. Despite the focus, many details still need to be elaborated for e.g. particular classes of use cases.

For the purpose of the DataBench project, the key (and the most concrete) part of the ontology are the indicators to be measured for each particular benchmark. The indicators are structures along the four main groups, each having subgroups and concrete values (to be extended for the benchmarks beyond the current scope of the DataBench project). The figure 11 depicts the main structure of the indicators as developed in WP1-WP4 first year deliverables.

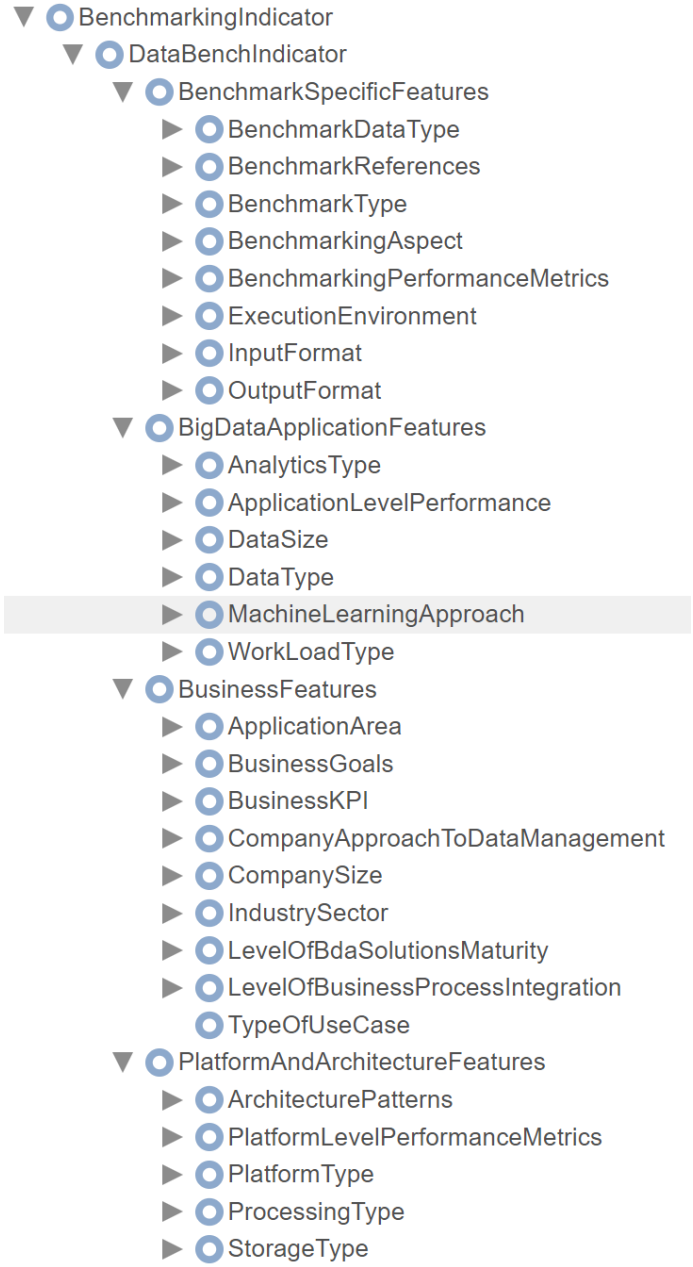

Figure 11 - The "BenchmarkIndicator" concept

Figure 11 – shows the "BenchmarkIndicator" concept structured into the four main groups of benchmarking indicators in the the DataBench ontology.

Each of the groups of indicators is further detailed into particular classes with corresponding values. As an example, the figure 12 is showing the class of "BigDataApplicationFeatures" indicators with corresponding values.

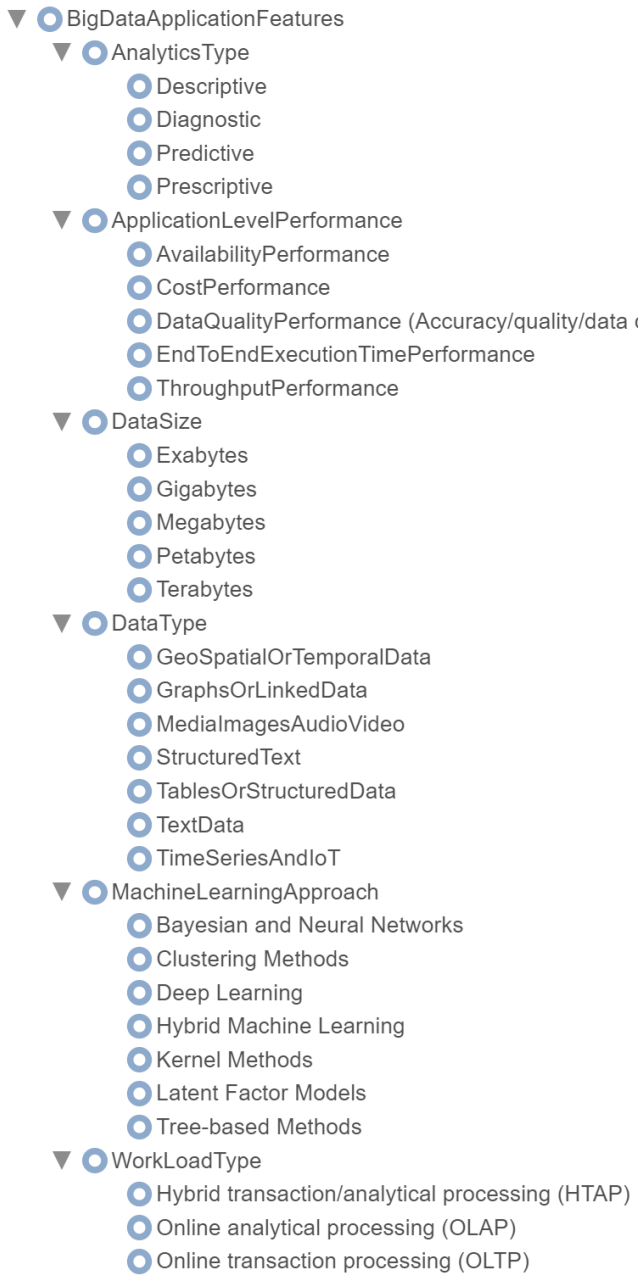

Figure 12 - The "BigDataApplicationFeatures" concept, indicators and values.

The part of the benchmarking module describing the data analytic part of the benchmarking ecosystem includes organizations dealing with benchmarking, their standardized benchmarks and connected to domains and particular use cases. As an example, figure 13 is showing the organizations dealing with benchmarking and more detailed elaboration for the TPC organization.

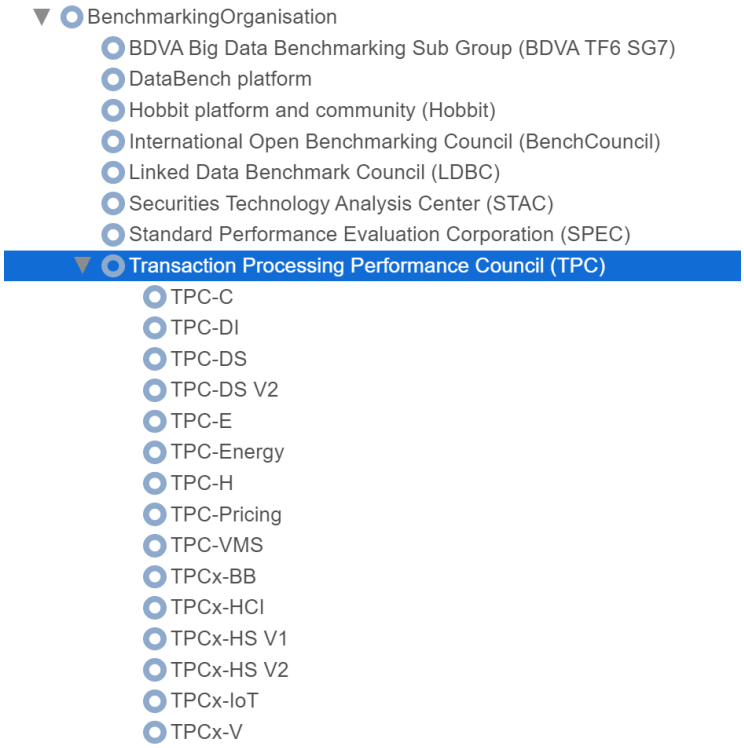

Figure 13 - The "Benchmarking Organisation" concept

Figure 13 – shows the "Benchmarking Organisation" concept with corresponding organizations with detailed view to TPC standardized benchmarks.

4.3 DataBench Ontology

As described in the section 4.2, the DataBench top level ontology structure consists of three main modules describing the benchmarking, analytic process, and data elements. Putting them together and connecting three modules at the appropriate concepts and via relations creates the DataBench ontology. The ontology could be observed from several angles. Maybe, the most illustrative side goes from the process concept "Benchmark" explaining good part of the rest of the ontology. Figure 14 shows how a typical benchmark is represented throughout the ontology connecting relevant concepts.

The ontology will serve as a basis for the DataBench Toolbox (WP3) to instantiate the concrete instances of the benchmark experiments. As continuation part of the work on WP5, the ontology will be used to store individual experiments to create the meta-learning model for various machine learning algorithms (as described in the section 5) with the purpose to create a landscape of the analytical part of the data analytic experimentation.

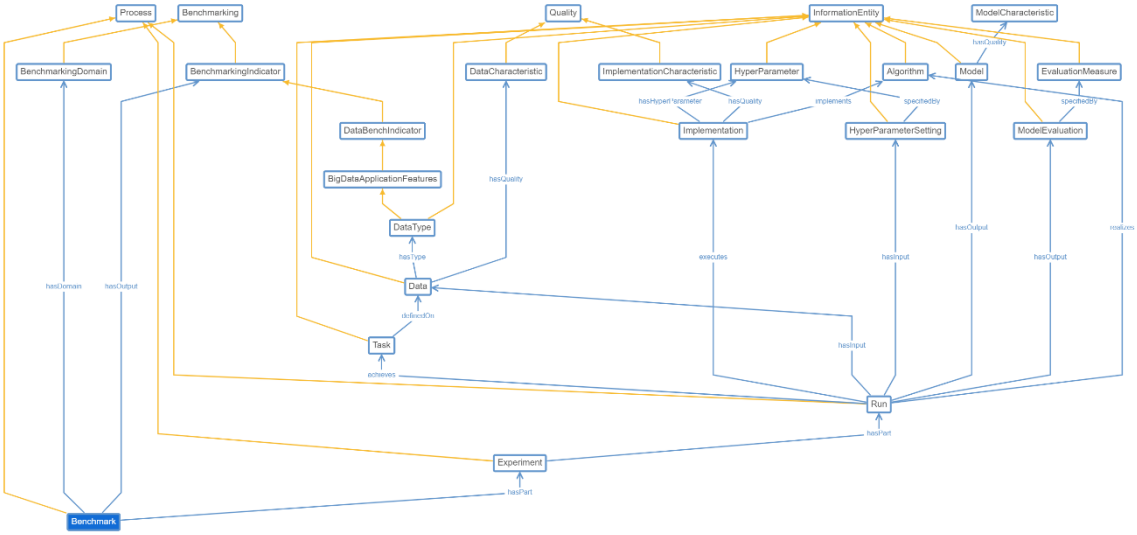

Figure 14 - The DataBench ontology

Figure 14 – depicts the DataBench ontology as observed from the "Benchmark" concepts, connecting all ontology modules to describe a benchmarking experiment.

The ontology can be downloaded in the XML/OWL format from the link <http://bit.ly/DataBenchOntology>. The link includes the latest revision of the file.

5. Towards Automatic Extraction of Indicators from Data

The central object of Machine Learning, Big Data and more broadly Data Science is a data set. Typical approach, when analyzing data, is to use the experience and 'feeling' of the data scientist – some types of data modalities have usual characteristics (such as text and images) while others have unknown distributions and require some prior investigation, which usually doesn't happen. A typical data scientist uses his/hers favorite data science tools and through an empirical test and error paradigm converges to better or worse result.

In WP5 we want to overcome such a bad practice and provide a tool which will take a target data as input and create a meta description of such dataset for the purpose of deeper understanding and improved decision in the follow-up steps. Such meta descriptions would include not just generic indicators but extract shallow (i.e. not as deep as a machine learning algorithm would reach) characteristics of the data. This includes describing probability distributions of the corresponding variables and insightful relations (through various statistical relational measures like correlation and beyond) between the variables. Those are typically never observed ahead of time by a data scientist and are only spotted (and not reported) by the analytic algorithm.

The major goal is not just to report characteristics of a data set, but rather link the extracted information with the properties of individual relevant analytic algorithms. As a result, the outcome would be a recommendation service where, given a data set, the system would propose which analytic algorithms would perform best and under what parametrization. As a general goal, we would like to create a 'landscape' of what kind of data performs well under what conditions and for what kind of tasks.

The area of data and algorithm characterization used to be known in Machine Learning as 'Meta Learning' which got lots of attention in the early years of Machine Learning and is not fully re-developed in the recent years in the time of Big Data and Deep Learning.

The actual work in WP5 will include approaches which were used in the past and propose new approaches relevant for the todays' setups and techniques. In particular, we expect contributions on the side of one pass algorithms of large datasets to extract useful characteristics from diverse datasets.

The following is a structured description of the dataset characterization approach. Note that a similar procedure will be designed for other segments related to data processing (including data storage, acquisition, pre-processing, interpretation visualization, architecture):

- **Input**: a *data-set* to be analysed
    - *Example data-sets*: sensor data (streaming, partial, noisy), text (traditional documents, Web, social media, news, multilingual), multi-media (audio/images/video), geospatial/spatiotemporal, and traditional structured data from relational databases
- **Processing**:
    - *An algorithm* having either one pass through the data or sampling the data source (in the case of large size or a stream of data)
    - The algorithm collects properties of the data-set including statistical properties of the variables and shallow relations between variables
- **Output**: *actionable characterization* in a form of a vector of meta indicators and corresponding tuneable similarity metrics:
    - The **indicators will structure the metrics characteristics along several key dimensions** to make
        - (a) different datasets/experiments comparable, and
        - (b) to provide an interpretable metrics landscape based on how data is being used.
    - Some **key dimensions** include: the aspects of storage, access, streaming, data-modality, integration, semantics, pre/post-processing, modelling, reasoning and visualization, anonymization/privacy, and legal/copyright.
    - We expect to be able to measure the characteristics of **few hundreds** (up-to thousand) different datasets.
    - Among others, the metrics should allow visualization and exploration.

The broader context, where the above data characterization will be used is to perform modelling to establish relationship between data characteristics and algorithmic performance and business indicators. The result will be an interpretable model allowing to interpret how data, algorithms/tools and business KPIs are connected and where are the better or worse scenarios to be used in practice. The approach to link different classes of KPIs will be based on collecting sufficient data for each of the classes and to create statistical

soft mappings among them. There is a risk in the cases where not enough data will be collected which we plan to mitigate with human interventions in a form of background knowledge (such as interpretable rules). The overall aim is to gain an interpretable correspondence between KPIs collected in diverse situations.

The sketch of the procedure is the following:

- **Input**: Dataset annotated with technical and business indicators.
    - Datasets, characterized and landscaped in the Task 5.1.
    - Dimensions impacting business decisions (Task 4.2) such as (a) scalability, (b) analytic task complexity, (c) technical usability, and (d) relevance.
- **Output**: Model/analytic mapping from dataset characteristics and methodology being used in the observed projects.
    - Datasets will be associated with tasks and possibly systems used.
    - The created model will statistically estimate mapping from a dataset and its associated tasks to challenges of how the data is being used in terms of selection of project infrastructure, methods, tools, user interfacing and legal challenges.

Summary

WP5 has the role in the project to use the data technology for the purpose to evaluate and validate Big Data benchmarking scenarios. The present document described three scenarios how WP5 will approach data benchmarking validation:

- Construction of the DataBench ontology to structure and systemize all the terms related to the Big Data benchmarking and to allow a level of reasoning over the collected data.
- Construction of the DataBench ontology and Knowledge Graph (DataBenchKG) to store all the data collected by the project into flexible schema graph database.
- Using analytics techniques to characterize data resources and relate them to algorithmic, tools and business KPIs for the purpose of recommending what should be used in particular Big Data scenario.

Bibliography

[1] Chen, Yanpei, "We Don't Know Enough to Make a Big Data Benchmark Suite" An Academia-Industry View, Unpublished paper presented at the Workshop on Big Data Benchmarking. May 2012, San Jose, CA.
https://amplab.cs.berkeley.edu/publication/we-dont-know-enough-to-make-a-big-data-benchmark-suite-an-academia-industry-view/

[2] Dan McCreary, "2018: The Year of Enterprise Knowledge Graphs" https://medium.com/@dmccreary/2018-the-year-of-enterprise-knowledge-graphs-66e868762b49 , Jan 2018

[3] Linked Open Data Cloud, https://lod-cloud.net/

[4] Sharan Vaswani, Meta Learning, http://www.cs.ubc.ca/labs/beta/Courses/CPSC532H-13/Slides/content-session-4-slides.pdf

[5] Vilalta, Ricardo, and Youssef Drissi. "A perspective view and survey of meta-learning." Artificial Intelligence Review 18.2 (2002): 77-95.

[6] Favio Vázquez, "Ontology and Data Science" https://towardsdatascience.com/ontology-and-data-science-45e916288cc5

[7] OWL – Web Ontology Language - https://www.w3.org/OWL/

[8] Protégé "A free, open-source ontology editor and framework for building intelligent systems" – https://protege.stanford.edu/

[9] Christiane Lemke, Marcin Budka, and Bogdan Gabrys: "Metalearning: a survey of trends and technologies", Artificial Intelligence Review 2015; 44(1): 117–130.

[10] ML Schema Core Specification - http://www.w3.org/2016/10/mls/ - Oct 17th 2016

[11] Data Catalog Vocabulary (DCAT) - Version 2 - https://www.w3.org/TR/vocab-dcat-2/ - W3C Proposed Recommendation, published on Nov 19th 2019