Describing Data Processing Pipelines in Scientific Publications for Big Data Injection

Sepideh Mesbah, Alessandro Bozzon, Christoph Lofi, Geert-Jan Houben
Delft University of Technology
Delft, the Netherlands
{s.mesbah, a.bozzon, c.lofi, g.j.p.m.houben}@tudelft.nl

ABSTRACT

The rise of Big Data analytics has been a disruptive game changer for many application domains, allowing the integration into domain-specific applications and systems of insights and knowledge extracted from external big data sets. The effective "injection" of external Big Data demands an understanding of the properties of available data sets, and expertise on the available and most suitable methods for data collection, enrichment and analysis. A prominent knowledge source is scientific literature, where data processing pipelines are described, discussed, and evaluated. Such knowledge is however not readily accessible, due to its distributed and unstructured nature. In this paper, we propose a novel ontology aimed at modeling properties of data processing pipelines, and their related artifacts, as described in scientific publications. The ontology is the result of a requirement analysis that involved experts from both academia and industry. We showcase the effectiveness of our ontology by manually applying it to a collection of Big Data related publications, thus paving the way for future work on more informed Big Data injection workflows.

Categories and Subject Descriptors

Computing methodologies [**Ontology engineering**]; Information systems [**Digital libraries and archives**]

Keywords

Big Data Injection, Information extraction from Scientific publications, Ontology Engineering, Data Processing

1. INTRODUCTION

Big Data analytics contributed to improvements in the state-of-the-art of several domains. Domain-specific data processing workflows (or "pipelines") facilitate the integration (at scale) of rich and meaningful knowledge mined from third-party, domain-agnostic Big Data sources, thus often opening the field for before unseen innovation.

In this respect, social media data represents a common yet successful example; the collection and analysis of users' activities enabled novel studies in: urban planning (e.g. activity spaces analysis through points-of-interest mining [11]);marketing (e.g. analysis of consumer-brand relationships [10]); public health care (e.g. real-time monitoring of diseases diffusion [2]); or pharmacovigilance (e.g. discovery of adverse effects of drugs [20]).

We refer to this powerful practice of integrating external (big) data sources, by means of processing pipelines, and to extend and supplement the power of an information system for achieving new goals, as "Big Data injection". Efficient and effective injection of Big Data is not a straightforward activity: to build novel solutions, Big Data practitioners and data scientists are required to have a deep understanding of the properties and limitations of the available data sources; of existing data processing pipelines devoted to the collection, enrichment and analysis of data; and of their respective implementations. We argue that the lack of suitable models and tools able to encode and collect such knowledge is one of the main roadblocks for a more principled and widespread adoption of Big Data injection.

In this work, we focus on (scientific) publications as a primary source of knowledge related to big data sources and data processing pipelines. Example of knowledge commonly contained in publications include: 1) the properties of big data sets of interest (e.g. size, sparseness, diversity, or bias); 2) the properties and limitations of related data processing techniques (e.g. complexity, accuracy); and 3) the properties of software and tools for data processing (e.g. run-time performance). Unfortunately, this rich knowledge is not readily accessible, as it is distributed across a vast repository of unstructured natural-language documents. To the best of our knowledge, no state-of-the-art computer system is currently able to provide an answer to the following query: *"Find the methods for POI (Point-Of-Interest) recommendation on 4Square data having a precision no lower than 10% from the state of the art"*. A first step towards the creation of a system able to answer such query is the availability of a knowledge representation models (e.g. an ontology) able to capture relevant properties of data sources, methods, and software that are relevant for Big Data Injection purposes.

Previous work tackled in several ways the representation, with ontologies, of some aspects of data sources and data processing pipelines in scientific publications [1][12]. However, to the best of our knowledge, no existing ontology is able to capture all the classes, properties, and relationships needed in order to answer the aforementioned query.

In this work, we propose a novel ontology called DMS [1] (Dataset, Method, and Software) able to encode and describe properties of data processing pipelines for Big Data injection in a machine-readable way. We elaborate on the the requirement elicitation process that lead to the creation of the DMS ontology; the process included an expert study, and an extended analysis of the state-of-the-art of related ontologies. We showcase the effectiveness of our ontology by manually annotating a collection of Big Data related publications, showing that indeed all relevant information can be captured. Finally, we outline and discuss our vision in how this ontology will be integrated into a larger ecosystem including sophisticated information extraction and reasoning in order to allow for semantically rich queries supporting practitioners with designing Big Data injection workflows.

2. REQUIREMENT ELICITATION

In this section, we will elaborate on the activities that led to the elicitation of the requirements that concluded in the current version of the DMS ontology.

2.1 Methodology

The design of the DMS ontology for data processing pipelines description has been performed according to the *Methondology* guidelines presented in [5].

To scope the requirement elicitation activity, we identified a domain of interest relevant for Big Data Injection, namely "social media data analysis". As also argued in the introduction, this domain finds widespread application and attracted considerable academic and industrial interest. In addition, data processing pipelines for social media data feature a full range of activities: from data set creation (e.g. crawling) and analysis, to the design of novel data enrichment methods (e.g. semantics of locations); to the adoption of existing methods and software (e.g. LIWC, Twitter API).

To capture the perspectives of both producers and consumers of publications related to big data sources and pipelines, the requirement elicitation process involved two classes of relevant actors: *Data Science Practitioners*, and *Data Scientists* from academia. We engaged with practitioners to discuss and identify relevant use cases for Big Data Injection, in the domain of social media data analytics. Scientists were interviewed in order to collect knowledge about the information that could be found in scientific publications, and that could be considered relevant for big data injection purposes from a scientific point of view.

2.2 Identification of Industrial Case Studies

We interacted with practitioners from the *Data Science & Analytics* unit of *Capgemini Netherlands*. Being involved with tasks related to big data and big data processing, the unit is a relevant and informed party for investigation. After an initial brainstorming on relevant use cases, we focused on the *Searching* use case, i.e. the task of retrieving, from collections of scientific publications, relevant information about available data sources and data processing pipelines. By means of semi-structured interviews, we derived a set of information needs (queries) related to the discovery of knowledge about data set, method and software from (scientific) publications. Examples of derived information needs include:

Searching for data sets: Researchers and practitioners are often looking for innovative applications of known data sets to new applications. Here, a typical query would be: *Find the available Web data sets that contain demographic information, but that have never been used in our organisation to study cultural differences across Dutch cities.*

Searching for methods: Researchers and practitioners often have to decide which methods will satisfy the domain requirements with respect to pre-defined metrics, e.g. precision and reliability. A typical sample query would be: *Find the method for POI recommendation based on Matrix Factorisation that features the best AUC metric in literature.*

Searching for software: On a similar note, researchers and practitioners are often interested in comparing software implementing aforementioned methods with respect to properties like performance or scalability. A typical example query would be: *Find the software used to tag objects in the images of the social media data (e.g. Instagram) with a precision within 20% from the state of art but with image annotation time lower than 100 milliseconds*

These examples of information needs clearly hint to three core functional requirements: 1) the ability to extract from a source of knowledge (e.g. a publication), preferably in an automated way, the information nuggets that contain relevant information about data sets and data processing pipelines; 2) the ability to link such information nuggets across different resources (e.g. publications, public or legacy databases); and 3) the ability to reason upon a body of knowledge, so as to infer properties that are not directly encoded in the original resource (e.g. the property of being "state-of-the-art").

2.3 Expert Analysis of Scientific Publications

For a system to satisfy the requirements 2) and 3) described above, the information nuggets contained in a publication must be first identified. Their identification allows the distillation of a set of concepts, properties, and relationships, that will constitute the main elements of the structured representation of the information contained in publications.

To this end, the authors interviewed three data scientists operating in the field of database systems, information retrieval, and software engineering. The three experts operate in our faculty, and were selected based on their academic and industrial experience with data processing pipelines. We selected five relevant and recent publications [4, 9, 13, 16, 22], and we asked the three experts to annotate them. This selection focuses on papers with a complete coverage of the respective data processing pipeline and its context in the domain of interest. The annotation tasks required the highlighting – with different colours – of paragraphs (or sentences) containing information relevant information about data sets, data processing pipelines, and the methods and software therein developed or employed. The scientists were also required to complement the annotation with a free-text description of the relevant attributes contained in the text (e.g. size of data set, parameter of used methods, link to software). We manually processed the experts annotations. We observed overlaps between highlighted paragraphs, as well as some differences in terms of the level of details in free-text annotations. We interpreted overlapping highlights as a signal of relevance for the annotated text. Also, we extracted relevant terms from the free-text annotations, and resolved synonymity among terms.

[1] Supporting website: `http://www.wis.ewi.tudelft.nl/DMS_SWM2017`

3. ONTOLOGY DESIGN

The requirements elicitation process led to the identification of four core concepts relevant for describing knowledge about big data data sources and processing pipelines in *publications*: *data sets*, *methods*, *software implementations*, and *experiments*.

While the first three concepts are evidently relevant, the fourth deserves clarification. When considering the example queries provided by the industry practitioners in Section 2.2, it clearly emerges the strict relationship that exists between several aspects of data processing (e.g. pipelines) and the experimental set-up described in a research paper. That is, in order to realise a certain research `objective`, an `experiment` is instrumented where a specific `combination` of methods is applied to a data set as part of a data processing pipeline, thus achieving a specific `performance` and `result` in that `context`.

In this section, we describe the final conceptualization of the DMS ontology, based on a term-extraction process (Section 3.1) and also by studying and integrating existing ontologies (see section 3.2) that are related, but not sufficiently expressive to cover the needed concepts, properties and relationships. Section 3.3 describes the resulting DMS ontology.

3.1 Term extraction

Term extraction is a central step in ontology engineering, in which the key concepts of the ontology and their characteristics are identified. We base our term extraction on the expert interviews, and their annotations as discussed in the previous section. Figure 1 depicts an excerpt from [4], showing rhetorical phrases annotated by one of the experts. These phrases encode characteristics of the data set used by the publication the excerpt was taken from (highlighted in blue), for instance: where to obtain the dataset, its size, and its temporal coverage. The goal of term extraction is to identify ontology terms and concepts which can explicitly encode the desired information in such phrases, for a large library of documents.

In the following, we focus on the *properties* of the core concepts covered by DMS. We started by collecting all raw terms related to properties of the concepts mentioned by the experts during the interview. For the annotations (as in Figure 1), we assigned a label from an uncontrolled vocabulary to all highlighted rhetorical phrases which best describes the encoded property. As a next step, we then manually grouped all resulting terms and labels with respect to their semantics, and finally subsumed each group into one property as shown below. During this process, we identified some additional concepts that are important to describe how data sets, methods, and software interact as parts of a Big Data injection workflow (as for example, some method is *applied* to a data set in an *experiment* which has a very specific *objective*.) We discuss these meta- and auxiliary concepts in section 3.3.

Data sets used or created in a publication. Data sets can be described by means of:

- The schema properties of the data set, such as the set of attributes (i.e actual data stored into a file, like a Json file which contains Twitter data with date, time, user, and content.)
- Quantitative properties of the data set, such as the size, and descriptive statistics like sparsity or skew. In our annotated publications, these properties are often encoded in tables.
- Temporal and Spatial properties of the data set (often found in text, e.g. ”data gathered between October 2009 and September 2011 from the French region in Switzerland [15]”).
- The application of the data set (also usually found in text, e.g. ”tracking Twitter for public health”).
- The scope of the data set (e.g. social media data, census data).
- The URL linking to the location of the data set (this is often found in text, footnotes, or references).
- The license (e.g. ”public domain”)

Methods, i.e. algorithms (novel or pre-existing) used to create, enrich, or analyse a data set. Methods can be described by means of:

- The parameters (often found in text, e.g. ”we used 10 fold cross-validation”).
- The data sets and parameters used or created by the method.
- Reference to an existing method (e.g. reference to another paper, references to implementing software)
- The application of the method (e.g. ”Lasso regularized regression was employed to modeling brand personality”)
- The result of the employed method (e.g.”the model predicted R^2 values as high as 0.67”, which is also usually found in form of tables or figures).

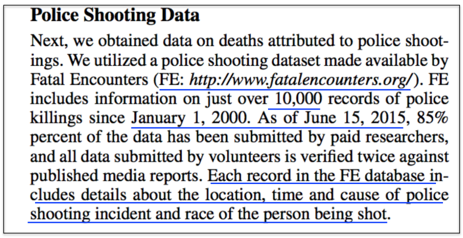
Software, i.e. computer tools employed to support the creation or the processing of data. Software can be can be described by means of:

- The result produced by the software.
- The license (e.g. ”public domain”).
- The application of the software (e.g. ”emotional expression measures were computed using LIWC”).
- The URL linking to the download location of the software.
- The programming language used.
- The performance of the software in the context of the experiments.

Our findings are summarised in table 1. The requirements elicitation activity highlights the need for the representation of data set properties, along with provenance information with respect to their creation and processing, and their relationship with methods and software organised in data processing pipelines designed for specific usage contexts.

3.2 Reuse of existing ontologies

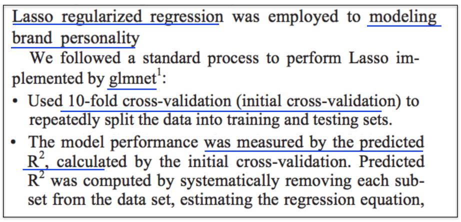
One design goal of DMS is to rely on the lessons-learned of established ontologies, and reuse their vocabulary whenever possible. Therefore, we provide an overview of the current state-of-the-art of related ontologies with a focus on the previously identified requirements in Table 1, and discuss them with respect to the three core concepts of publications, methods, data sets, and software in the following.



Police Shooting Data

Next, we obtained data on deaths attributed to police shootings. We utilized a police shooting dataset made available by Fatal Encounters (FE: *http://www.fatalencounters.org/*). FE includes information on just over 10,000 records of police killings since January 1, 2000. As of June 15, 2015, 85% percent of the data has been submitted by paid researchers, and all data submitted by volunteers is verified twice against published media reports. Each record in the FE database includes details about the location, time and cause of police shooting incident and race of the person being shot.

Figure 1: Text excerpt with mentions of data source attributes highlighted.



Lasso regularized regression was employed to modeling brand personality

We followed a standard process to perform Lasso implemented by glmnet[1]:

- Used 10-fold cross-validation (initial cross-validation) to repeatedly split the data into training and testing sets.
- The model performance was measured by the predicted R^2, calculated by the initial cross-validation. Predicted R^2 was computed by systematically removing each subset from the data set, estimating the regression equation,

Figure 2: Example text containing information about a method and a software.

	DMS [2]	DOCO[3]	DEO[3]	Disco [1]	CiTo [18]	OntoSoft [6]
Describing Data sets						
Variables-Data files	+	-	-	+	-	-
Quantitative properties	+	-	-	+	-	-
Temporal and Spatial	+		-	+	-	-
Scope	+	-	-	+	-	-
License	+	-	-	+	-	-
Link to location	+	-	-	-	-	-
Describing methods						
Methods	+	-	+	-	-	-
Method parameters	+	-	-	-	-	-
Results	+	-	+	-	-	-
Application	+	-	-	-	-	-
Citation	+	-	-	-	+	-
Describing software						
Programming language	+	-	-	-	-	+
Average runtime	+	-	-	-	-	+
Describing Experiments						
Objective	+	-	-	-	-	-
Research Questions	+	-	-	-	-	-
Figures-table	+	+	-	-	-	-

Table 1: Comparison established ontologies with respect to our three core-topics, and publication specific meta properties. Plus sign (+): The property has been covered by the ontology, Minus sign (-): The property has not been covered by the ontology. ([2]Our proposed ontology, [3]http://purl.org/spar/deo) .

Describing scientific publications and methods. In this work, we focus on properties of big data sources and processing pipelines as extracted from scientific publications. Many aspects of the nature of the overall data processing pipeline are described in the rhetoric of the research publication itself (like in the motivation, abstract, or discussion). Several ontologies exist for describing structural properties (e.g title, sections, header, etc.) and rhetorical elements (e.g contribution, results, figures, tables and etc) of scientific publications. The Semantic Publishing and Referencing Ontologies (SPAR Ontologies)[4], is one of the first attempts to describe different aspects of semantic publishing and referencing in a machine-readable format. SPAR consists of 13 OWL2 DL ontology modules. For the sake of brevity, we only describe the ones (Doco, Deo and Cito) that are related to our properties of interest. The Document Components Ontology (Doco) [3] provides a structured vocabulary for both structural (e.g. block, inline, paragraph, section, chapter) and rhetorical (e.g. introduction, discussion, table, reference list, figure, appendix) components of the paper. Doco imports Discourse Elements Ontology (Deo)[5] which provides a vocabulary for rhetorical elements within documents, including *methods* and results. DoCo and Deo both complement eachother. Ruiz-Iniesta in [19] reviewed the scholarly document ontologies and suggested Doco and Deo for describing the structural and rhetorical elements of the publications, and the Citation Typing Ontology (Cito) [18] for describing the citation acts between the scientific publications (e.g. cito:cites, cito:extends, cito:isDescribedBy).

Notice that the mentioned ontologies do not directly address properties of data sets or software.

Ontologies for describing data sets. The DDI-RDF Discovery Vocabulary (Disco) [1] and RDF-Data Cube(q) vocabulary[6] provide a description of the schema of a data set as well as its quantitative properties. Here, the RDF-Data Cube(q) vocabulary focuses specifically on aggregated data stored in data cubes, allowing to describe the cube's structure as well as the representation of the contained data. In contrast, the Discovery Vocabulary covers the description of raw data, but is not concerned with its representation. It also focuses mainly on file-based data sets. The Disco vocabulary also makes use of DCMI Metadata Terms [7] for describing properties like temporal and spatial extend, or information like licenses. We therefore reuse the Disco vocabulary in DMS, to describe the schema and properties (including temporal, spatial, and license-related properties) of data sets.

[4] http://purl.org/spar

[5] http://www.sparontologies.net/ontologies/deo/source.html

[6] https://www.w3.org/TR/vocab-data-cube/

[7] http://dublincore.org/documents/dcmi-terms/

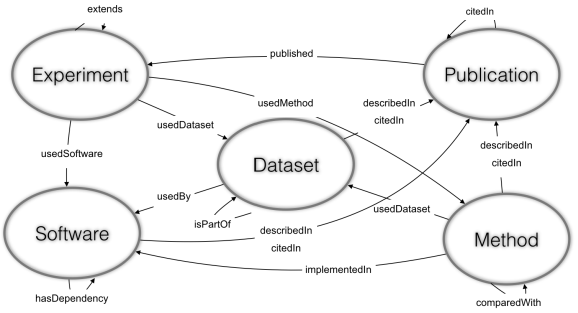


Figure 3: Abstract view of the DMS ontology.

Ontologies for describing software: sciObjCS ontology [17] describes scientific objects (e.g. tools) along with their categories and creators. The OntoSoft ontology [6] is an OWL-based ontology for describing meta-data of scientific software. The ontology supports scientist to identify the software, and allows to cover many properties related to installing or running it. We deem the OntoSoft ontology complete; therefore, we exploit its classes and properties in DMS, to capture attributes such as dependencies, programming language, average runtime, etc.

Discussion. From our analysis of the state-of-the-art, we can conclude that existing ontologies already cover a subset of the identified requirements, but they all fall short covering the whole picture. Therefore, we aim at bridging this gap by combining relevant aspects of those ontologies, extending them with new concepts and properties. We also especially focus on the non-trivial relationships necessary for describing a pipeline, connecting data sets, methods, and software for a targeted usage context.

3.3 Ontology conceptualization

In this section, we will outline our conceptualisation of the DMS ontology. In addition to the ontologies listed in Table 1, we build upon the PROV-O [8] ontology by extending the Entity PROV-O class for each of the classes of the DMS ontology that benefit from withholding provenance information (e.g. the creator, the location of a data set, etc). We further partially reused SKOS [9] ontology to make use of the taxonomy concepts (e.g. broader, narrower).

Figure 3 provides a high-level, abstract view of the DMS ontology. The core concepts of our ontology are *data sets*, *methods*, and *software*. *Publications* implicitly describe pipelines, usually as parts of different *experiments*.

The DMS ontology has been implemented using OWL 2 DL, and consists of 10 classes and 30 properties. Table 2 summarises the novel classes and properties included in DMS.

In Figure 4, as an example, we zoom in the set of properties describing a data set, which is based on the *logicalDataset* concept of the *disco* ontology. The general properties of a data set cover the creator, licence, scope, link to location etc. Each *logicalDataset* has some data files (*disco:datafile*), and multiple logical data sets form the final data set. For example, one experiment might refer to a JSON file that contains a specific set of Twitter messages. We can distinguish the ground truth data set used in the experiment, with the *dms:isGroundTruthData* property. The variables (i.e., schema attributes) contained in the data set (e.g longitude, latitude of the tweets) can be defined using *disco:variable*. Dependent variables used in the experiment can be distinguished using the *dms:isDependentVariable*. Each data set has a scope (*dms:scope*) (e.g social media) which can be linked to a concept (skos:concept). It also includes the description of the temporal and spatial coverage of the dataset, which can be described using *dctersm:temporal* and *dctersm:spatial*. The statistical properties of the data set can be described using the *disco:DescriptiveStatistics* concept. The data files and attributes used for each statistics can be described with the *disco:statisticsDatafile* and *disco:statisticsVariable* property.

Dataset	Classes	Scope, Application
	Properties	hasApplication , hasScope, hasStatisticMeasurement, isDependentVariable isGroundTruth
Method	Classes	MethodImplementation, Parameter, Application, AcceptanceRange,
	Properties	createdDataset, createdVariable, usedDataset, usedVariable, hasImplementation, implementedIn, referenceObjective, hasAcceptanceRange produced, comparedWith, hasEndRange, hasStartRange, measurementType
Sofware	Classes	SoftwareConfiguration, Application
	Properties	createdDataset, createdVariable, usedDataset, usedVariable, referenceObjective, produced
Experiment	Classes	Publication, Experiment, Objective, ResearchQuestion
	Properties	describesExperiment, usedMethod, usedSoftware, hasFigure, hasTable, hasConfiguration,relatedTo, isSubGoalOf, isSubResearchQuestionOf, hasObjective, hasResearchQuestion, hasCaption,

Table 2: Summary of the novel classes and properties included in DMS. Some classes and properties are related both to the Method and the Software class

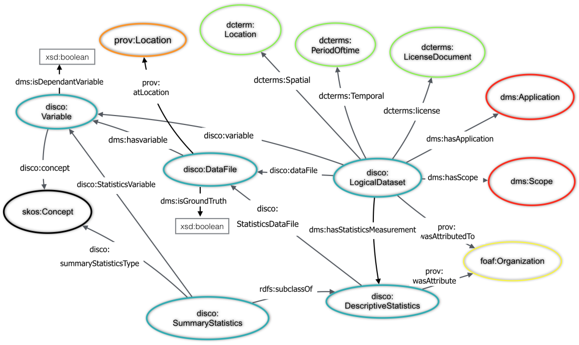


Figure 4: The ovals with different colours used in the figure are an indicator of different ontologies.

Figure 5 focuses on the parts of DMS that describe the link between data sets, methods, and software - this implicitly encodes the overall data processing pipeline discussed in a publication in the context of an experiment (as each experiment in the Big Data injection domain is usually sequence of applying methods to different data set, and assessing the result quality).

[8] https://www.w3.org/TR/prov-o/
[9] http://www.w3.org/2008/05/skos

Here, an experiment has an objective (dms:objective) and uses data sets (dms:usedDataset), and methods (dms:usedMethod) as provided by software (dms:usedSoftware). In each experiment, different implementations or configurations of a method or software can be used, motivating *Method-Implementation* concept (dms:MethodImplementation ⊂ deo:Methods). Since the *dms:Application* class, likewise the *dms:Scope* class is a concept, broader concept like ”recommendation” can be defined for it. Each application can be linked to the reference sub goal of the overall objective.

The dms:MethodImplementation, is either a new method described in the paper or an existing one (i.e. referenced) which can be used both for the creation (dms:createdDatset) or the analysis of a dataset. For instance, as shown in Figure 6, a method by using a datafile (dms:usedDataset), some variables (dms:usedVariable) of the dataset, and having some parameters with an acceptance range, it can produce (dms:produced) a result (e.g precision 70%) with a measurement type (e.g. precision).

This is the same for the dms:softwareConfiguration (⊂ ontosoft:Software) class, which can be used both for the creation or the analysis of a dataset.

4. VALIDATION BY APPLICATION

In this section we validate the suitability of DMS by a) manually annotating ten publications related to Social Media Big Data Injection, and b) providing example SPARQL queries to showcase that DMS can already satisfy many information demands identified by our industry practitioners in section 2 - even without having complex reasoning capabilities in place which will be provided by future implementations.

4.1 Annotation of Scientific Publications

We manually annotated 10 papers [22, 4, 9, 13, 16, 23, 15, 14, 8, 21] in the field of social media analytics to show that our ontology is indeed able to capture the relevant properties of data processing pipelines. A public SPARQL endpoint to the RDF encoding resulting from this annotation is freely accessible[10].

Listing 1 is a sample RDF representation of the annotation in Figure 1: the *Police Shooting Data* has some schema attributes (called `variables` in accordance to the vocabulary of the DISCO ontology which we imported for this purpose) such as the cause of shooting incident, information on the victims liek age, gender, or race, and also the time and location of the shooting. 10,000 incident records have been collected during the period between 01/01/2000 to 15/06/2015, and the URL linking to location of the data set is `http://www.fatalencounters.org`.

```
prefix ns2:   <http://purl.org/dc/terms/> .
prefix ns1:   <http://www.w3.org/ns/prov#> .
prefix ns4:   <http://purl.org/dc/elements/1.1/> .
prefix ns3:   <http://rdf-vocabulary.ddialliance.org/discovery#> .
prefix ns5:   <https://github.com/mesbahs/DMS/blob/master/dms.owl#> .

[ a                ns3:DataFile ;
  ns4:title        "Police Shooting Data" ;
  ns2:temporal     "01/01/2000-15/06/2015" ;
  ns3:caseQuantity 10000 ;
  ns5:hasVariable     [ a         ns3:Variable ;
                     ns1:value
                     "cause of police shooting incident" ] ;
  ns5:hasVariable     [ a         ns3:Variable ;
                     ns1:value  "time." ] ;
  ns5:hasVariable     [ a         ns3:Variable ;
                     ns1:value  "location" ] ;
  ns5:hasVariable     [ a         ns3:Variable ;
                     ns1:value
                     " race of the person being shot"  ] ;
  ns1:atLocation    "http://www.fatalencounters.org/"
] .
```

Listing 1: RDF representation of Figure 1.

In Figure 2, another example annotation is shown (from [22]). The different colors of the highlights represent information on methods, their application, used software and parameter, and the result of the overall pipeline. The resulting RDF is shown in listing 2.

```
prefix ns2:   <https://github.com/mesbahs/DMS/blob/master/dms.owl#> .
prefix ns1:   <http://www.w3.org/ns/prov#> .
prefix ns3:   <http://purl.org/dc/elements/1.1/> .

[ a                   ns2:methodImplementation ;
  ns3:title           "Lasso regularized regression" ;
  ns2:hasApplication [ a   ns2:Application ;
                        ns3:title  "modeling brand personality" ] ;
  ns2:hasParameter [ a  ns2:Parameter ;
                    ns1:value  "10-fold cross-validation"  ] ;
  ns2:produced [ a <http://purl.org/spar/deo#Results> ;
                    ns2:measurementType  "predicted R2 "   ] ;
  ns2:implementedIn [ a  ns2:softwareConfiguration ;
                    ns3:title      "glmnet"    ;]
] .
```

Listing 2: RDF representation of Figure 2.

After manually annotating the 10 papers, we found that in general the ontology was able to cover the required properties related to data processing pipeline. As expected, some DMS concepts and properties were used more frequentyl than others, such as: Application, MethodImplementation, SoftwareConfiguration, produced, hasObjective, describesExperiment, usedDataset, usedMethod, usedSoftware, hasApplication et. On the other hand, some concepts were rarely used such as Parameter, AcceptanceRange, ResearchQuestion, Scope, etc.

4.2 Use Case Queries

In this section we will describe a sample query which the DMS ontology was designed to support. We envision that by populating the DMS ontology we are able to answer queries like the following examples:

Find the methods that can rank POI recommendation with a precision no lower than the state of the art

In this case, its related SPARQL query is as follows:

```
SELECT  ?method ?resvalue
    WHERE {
        ?method a dms:MethodImplementation;
            dms:produced ?result;
            dms:hasApplication ?application.
        ?application rdf:type dms:Application;
            disco:concept ?skosConcept.
            ?skosConcept rdf:type skos:Concept;
            skos:notation ?notation;
            FILTER (regex(?notation,"poi recommendation","i")).
        ?result rdf:type deo:Results;
            prov:value ?resvalue;
            Filter(?resvalue>= ?sota).
        ?result dms:measurementType ?type;
        FILTER(regex(?type,"precision","i")).

```

[10] http://www.wis.ewi.tudelft.nl/DMS_SWM2017

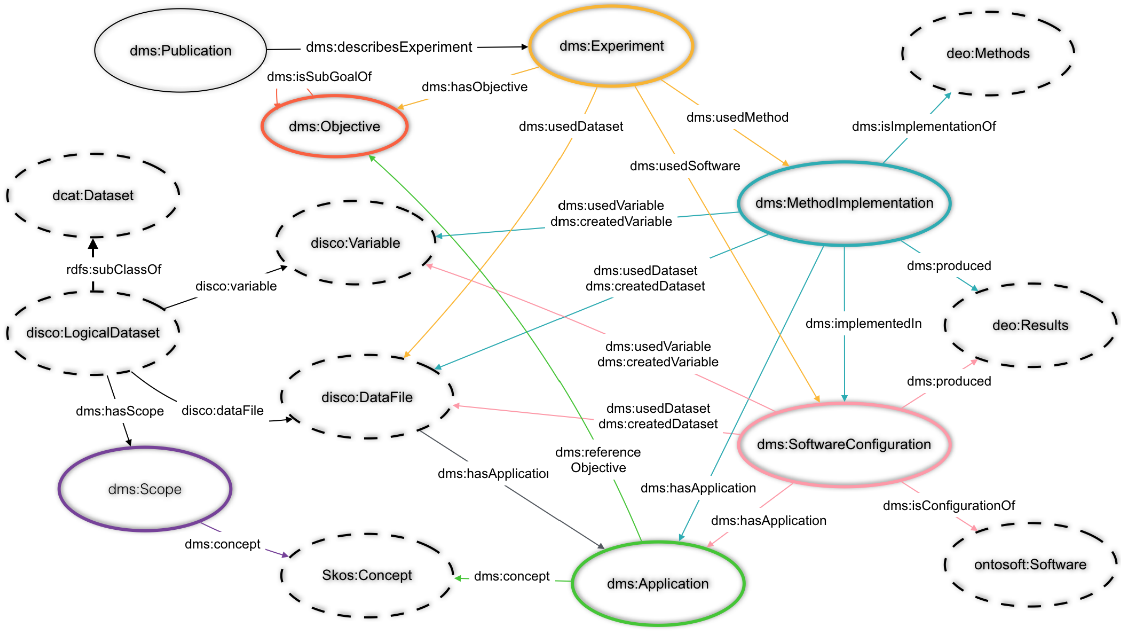


Figure 5: Linking Datasets, Methods and Software. The ovals with different colors used in the figure are an indicator of different classes that we defined, and the dashed ovals are an indicator of old classes that were defined by the existing ontologies.

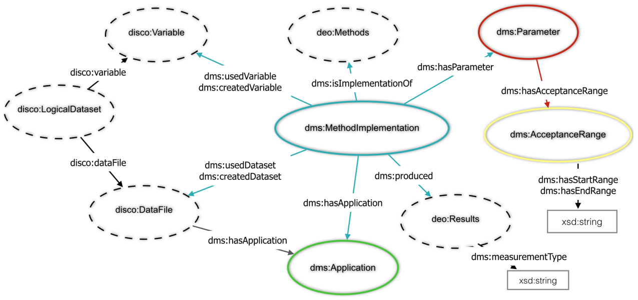


Figure 6: A portion of ontology describing the method

```
            { SELECT  ?sota
            WHERE {
               ?method a dms:MethodImplementation;
                   dms:produced ?result;
                   dms:hasApplication ?application.
                ?application disco:concept ?skosConcept.
                   ?skosConcept  skos:notation ?notation;
                   FILTER (regex(?notation,"poi recommendation","i")).
                ?result rdf:type deo:Results; prov:value ?sota;
                   dms:measurementType ?type;
                   FILTER(regex(?type,"precision","i")) }
               ORDER BY DESC(?sota) LIMIT 1}
                        }}}
```

This query returns all the methods that can rank POI recommendation, and retrieves the ones that have the highest precision among the state-of-the-art methods.

5. CONCLUSION & OUTLOOK

In this paper, we presented a novel ontology DMS for covering meta-data of data sets, methods, and software as parts of Big Data injection pipelines found in scientific publications. We have presented a rigid process for designing the ontology based on information needs of data science practitioners, and the input of seasoned academic data science researchers. We also reused existing ontologies and vocabularies whenever possible, thus limiting the overhead of adapting our new ontology, which finally covers 10 classes and 30 properties in OWL 2 DL. Finally, we validated the ontology by using it to annotate ten publications from the area of Social Media injection, and showing SPARQL queries which can indeed cover the information need identified by the practitioners in the requirement elicitation phase.

In conclusion, one of the most dominant use-cases for Big Data injection is the field of social media analytics, and thus the publications and experts we used for eliciting the requirements and evaluating the effectiveness of DMS are rooted in that field. As a result, we believe that DMS is able to cover publications in that area well, but might need additional considerations when transferred to other domains.

Also, we aimed at a more generalized conceptualisation of the ontology due to the diversity of methods, data sets, measurement types, applications of methods, etc. While the chosen OWL 2 DL knowledge representation formalism would have allowed for a more fine-grained model also including for example cardinalities or complex subclass taxonomies, we refrained from doing so with hindsight on the future usage of this ontology in a (semi-)automated information system (for example, the property ”measurement type” could be further specialized into ”precision”, ”recall”, etc - but we felt that this would unnecessarily complicate the ontology. This argumentation is also in line with the minimal encoding bias and extendability principle guidelines outlined in [7].)

In our future work, we will focus on realizing a larger ecosystem where the DMS ontology is semi-automatically populated using publications from digital library backend, thus building a rich knowledge repository of data processing pipeline meta-data which can serve as a nucleus for fostering future research on Big Data injection. This endeavour requires identifying rhetoric mentions of the properties and concepts covered in DMS in publications, not unlike the annotation task performed by our experts in the requirement elicitation step of this paper. This will likely motivate an expansion of DMS to also cover such rhetorical mentions on higher level of granularity which will be designed for human consumption, or as input for later processing steps (e.g. a natural language description of data set properties instead of a fine-grained explicit notion of the properties as used in this work.) From there, we plan on developing specialized extractors to infer the actual property values from rhetorical mentions wherever possible, or falling back to crowd-sourcing for harder cases. A second major challenge is realizing the reasoning capabilities necessary to support the queries identified during requirements analysis more effectively. For example, in the previous section, we implemented the notion of "current state-of-the-art" manually in SPARQL while in future versions of the system, such concepts should be usable without explicit definition.

6. REFERENCES

[1] T. Bosch, R. Cyganiak, A. Gregory, and J. Wackerow. Ddi-rdf discovery vocabulary: A metadata vocabulary for documenting research and survey data. In *LDOW*, 2013.

[2] L. E. Charles-Smith, T. L. Reynolds, M. A. Cameron, M. Conway, E. H. Lau, J. M. Olsen, J. A. Pavlin, M. Shigematsu, L. C. Streichert, K. J. Suda, et al. Using social media for actionable disease surveillance and outbreak management: A systematic literature review. *PloS one*, 10(10):e0139701, 2015.

[3] A. Constantin, S. Peroni, S. Pettifer, D. Shotton, and F. Vitali. The document components ontology (doco). *Semantic Web*, 7(2):167–181, 2016.

[4] M. De Choudhury, S. Jhaver, B. Sugar, and I. Weber. Social media participation in an activist movement for racial equality. In *Tenth International AAAI Conference on Web and Social Media*, 2016.

[5] M. Fernández-López, A. Gómez-Pérez, and N. Juristo. Methontology: from ontological art towards ontological engineering. 1997.

[6] Y. Gil, V. Ratnakar, and D. Garijo. Ontosoft: Capturing scientific software metadata. In *Proceedings of the 8th International Conference on Knowledge Capture*, page 32. ACM, 2015.

[7] T. R. Gruber. Toward principles for the design of ontologies used for knowledge sharing? *International journal of human-computer studies*, 43(5):907–928, 1995.

[8] T. Hu, H. Xiao, J. Luo, and T.-v. T. Nguyen. What the language you tweet says about your occupation. In *Tenth International AAAI Conference on Web and Social Media*, 2016.

[9] Y. Hu, S. Farnham, and K. Talamadupula. Predicting user engagement on twitter with real-world events. In *Proceedings of the International Conference on Weblogs and Social Media (ICWSM). AAAI*, 2015.

[10] S. Hudson, L. Huang, M. S. Roth, and T. J. Madden. The influence of social media interactions on consumer–brand relationships: A three-country study of brand perceptions and marketing behaviors. *International Journal of Research in Marketing*, 33(1):27–41, 2016.

[11] S. Jiang, A. Alves, F. Rodrigues, J. Ferreira, and F. C. Pereira. Mining point-of-interest data from social networks for urban land use classification and disaggregation. *Computers, Environment and Urban Systems*, 53:36–46, 2015.

[12] T. Kauppinen, A. Baglatzi, and C. Keßler. Linked science: interconnecting scientific assets. *Data Intensive Science. CRC Press, USA (forthcoming 2012)*, 2012.

[13] G. Le Falher, A. Gionis, and M. Mathioudakis. Where is the soho of rome? measures and algorithms for finding similar neighborhoods in cities. In *AAAI Conference on Web and Social Media*, 2015.

[14] T. Mitra, S. Counts, and J. W. Pennebaker. Understanding anti-vaccination attitudes in social media. In *Tenth International AAAI Conference on Web and Social Media*, 2016.

[15] S. A. Muhammad and K. Van Laerhoven. Duke: A solution for discovering neighborhood patterns in ego networks. In *The 9th International AAAI Conference on Web and Social Media (ICWSM), Oxford, England*, volume 5, page 2015, 2015.

[16] A. Olteanu, C. Castillo, F. Diaz, and S. Vieweg. Crisislex: A lexicon for collecting and filtering microblogged communications in crises. In *ICWSM*, 2014.

[17] F. Osborne, H. de Ribaupierre, and E. Motta. Techminer: Extracting technologies from academic publications. 2016.

[18] S. Peroni and D. Shotton. Fabio and cito: ontologies for describing bibliographic resources and citations. *Web Semantics: Science, Services and Agents on the World Wide Web*, 17:33–43, 2012.

[19] A. Ruiz-Iniesta and O. Corcho. A review of ontologies for describing scholarly and scientific documents. In *SePublica*, 2014.

[20] A. Sarker, R. Ginn, A. Nikfarjam, K. O'Connor, K. Smith, S. Jayaraman, T. Upadhaya, and G. Gonzalez. Utilizing social media data for pharmacovigilance: A review. *Journal of biomedical informatics*, 54:202–212, 2015.

[21] T. H. Silva, P. O. de Melo, J. Almeida, M. Musolesi, and A. Loureiro. You are what you eat (and drink): Identifying cultural boundaries by analyzing food & drink habits in foursquare. *arXiv preprint arXiv:1404.1009*, 2014.

[22] A. Xu, H. Liu, L. Gou, R. Akkiraju, J. Mahmud, V. Sinha, Y. Hu, and M. Qiao. Predicting perceived brand personality with social media. In *Tenth International AAAI Conference on Web and Social Media*, 2016.

[23] Z. Yin, Y. Chen, D. Fabbri, J. Sun, and B. Malin. # prayfordad: Learning the semantics behind why social media users disclose health information. In *Tenth International AAAI Conference on Web and Social Media*, 2016.